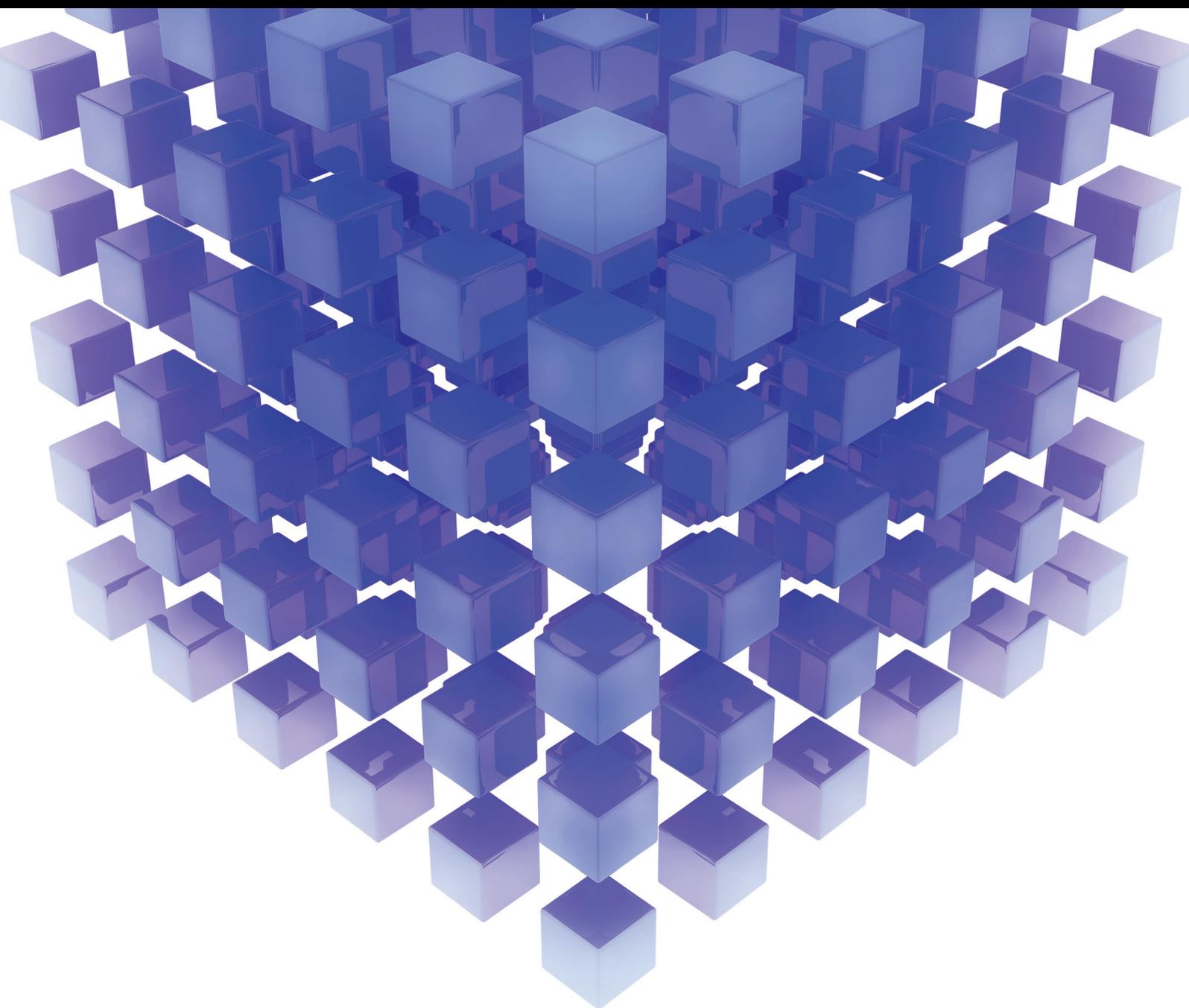


Mathematical Problems in Engineering

Computational Intelligence in Image Processing 2016

Guest Editors: Erik Cuevas, Daniel Zaldívar, Gonzalo Pajares, Marco Perez-Cisneros, and Raúl Rojas





**Computational Intelligence in
Image Processing 2016**

Mathematical Problems in Engineering

Computational Intelligence in Image Processing 2016

Guest Editors: Erik Cuevas, Daniel Zaldívar, Gonzalo Pajares,
Marco Perez-Cisneros, and Raúl Rojas



Copyright © 2016 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

- Mohamed Abd El Aziz, Egypt
Farid Abed-Meraim, France
José Ángel Acosta, Spain
Paolo Addresso, Italy
Claudia Adduce, Italy
Ramesh Agarwal, USA
Juan C. Agüero, Australia
Ricardo Aguilar-López, Mexico
Tarek Ahmed-Ali, France
Hamid Akbarzadeh, Canada
Muhammad N. Akram, Norway
Mohammad-Reza Alam, USA
Salvatore Alfonzetti, Italy
Francisco Alhama, Spain
Mohammad D. Aliyu, Canada
Juan A. Almendral, Spain
Lionel Amodeo, France
Sebastian Anita, Romania
Renata Archetti, Italy
Felice Arena, Italy
Sabri Arik, Turkey
Alessandro Arsie, USA
Eduardo Artioli, Italy
Fumihiro Ashida, Japan
Hassan Askari, Canada
Mohsen Asle Zaeem, USA
Romain Aubry, USA
Matteo Aureli, USA
Francesco Aymerich, Italy
Seungik Baek, USA
Khaled Bahlali, France
Laurent Bako, France
Stefan Balint, Romania
Alfonso Banos, Spain
Roberto Baratti, Italy
Martino Bardi, Italy
Azeddine Beghdadi, France
Denis Benasciutti, Italy
Abdel-Hakim Bendada, Canada
Ivano Benedetti, Italy
Elena Benvenuti, Italy
Jamal Berakdar, Germany
Michele Betti, Italy
Jean-Charles Beugnot, France
Simone Bianco, Italy
- Gennaro N. Bifulco, Italy
David Bigaud, France
Antonio Bilotta, Italy
Jonathan N. Blakely, USA
Paul Bogdan, USA
Alberto Borboni, Italy
Paolo Boscariol, Italy
Daniela Boso, Italy
Guillermo Botella-Juan, Spain
Abdel-Ouahab Boudraa, France
Fabio Bovenga, Italy
Francesco Braghin, Italy
Michael J. Brennan, UK
Maurizio Brocchini, Italy
Julien Bruchon, France
Michele Brun, Italy
Javier Buldú, Spain
Erik Burman, UK
Tito Busani, USA
Raquel Caballero-Águila, Spain
Pierfrancesco Cacciola, UK
Salvatore Caddemi, Italy
Jose E. Capilla, Spain
E. Javier Cara, Spain
Ana Carpio, Spain
Carmen Castillo, Spain
Inmaculada T. Castro, Spain
Gabriele Cazzulani, Italy
Luis Cea, Spain
Miguel E. Cerrolaza, Spain
M. Chadli, France
Gregory Chagnon, France
Ching-Ter Chang, Taiwan
Michael J. Chappell, UK
Kacem Chehdi, France
Peter N. Cheimets, USA
Chunlin Chen, China
Xinkai Chen, Japan
Francisco Chicano, Spain
Hung-Yuan Chung, Taiwan
Joaquim Ciurana, Spain
John D. Clayton, USA
Giuseppina Colicchio, Italy
Mario Cools, Belgium
Sara Coppola, Italy
- Jean-Pierre Corriou, France
Juan Carlos Cortés López, Spain
Carlo Cosentino, Italy
Paolo Crippa, Italy
Andrea Crivellini, Italy
Erik Cuevas, Mexico
Peter Dabnichki, Australia
Luca D'Acerno, Italy
Weizhong Dai, USA
Andrea Dall'Asta, Italy
Purushothaman Damodaran, USA
Farhang Daneshmand, Canada
Fabio De Angelis, Italy
Stefano de Miranda, Italy
Filippo de Monte, Italy
Maria do Rosário de Pinho, Portugal
Xavier Delorme, France
Frédéric Demoly, France
Luca Deseri, USA
Angelo Di Egidio, Italy
Ramón I. Diego, Spain
Yannis Dimakopoulos, Greece
Zhengtao Ding, UK
Mohamed Djemai, France
Manuel Doblaré, Spain
Alexandre B. Dolgui, France
Florent Duchaine, France
George S. Dulikravich, USA
Bogdan Dumitrescu, Romania
Horst Ecker, Austria
Ahmed El Hajjaji, France
Fouad Erchiqui, Canada
Anders Eriksson, Sweden
R. Emre Erkmen, Australia
Andrea L. Facci, Italy
Giovanni Falsone, Italy
Hua Fan, China
Yann Favennec, France
Giuseppe Fedele, Italy
Roberto Fedele, Italy
Jose R. Fernandez, Spain
Jesus M. Fernandez Oro, Spain
Eric Feulvarch, France
Simme Douwe Flapper, Netherlands
Thierry Floquet, France

Eric Florentin, France
Jose M. Framinan, Spain
Francesco Franco, Italy
Elisa Francomano, Italy
Mario L. Fravolini, Italy
Leonardo Freitas, UK
Tomonari Furukawa, USA
Mohamed Gadala, Canada
Matteo Gaeta, Italy
Mauro Gaggero, Italy
Zoran Gajic, Iraq
Erez Gal, Israel
Ugo Galvanetto, Italy
Akemi Gálvez, Spain
Rita Gamberini, Italy
Maria L. Gandarias, Spain
Arman Ganji, Canada
Xin-Lin Gao, USA
Zhong-Ke Gao, China
Giovanni Garcea, Italy
Fernando García, Spain
Jose M. Garcia-Aznar, Spain
Laura Gardini, Italy
Alessandro Gasparetto, Italy
Vincenzo Gattulli, Italy
Oleg V. Gendelman, Israel
Mergen H. Ghayesh, Australia
Agathoklis Giaralis, UK
Anna M. Gil-Lafuente, Spain
Hector Gómez, Spain
David González, Spain
Francisco Gordillo, Spain
Rama S. R. Gorla, USA
Oded Gottlieb, Israel
Nicolas Gourdain, France
Kannan Govindan, Denmark
Antoine Grall, France
Jason Gu, Canada
Federico Guarracino, Italy
José L. Guzmán, Spain
Quang Phuc Ha, Australia
Masoud Hajarian, Iran
Frédéric Hamelin, France
Zhen-Lai Han, China
Thomas Hanne, Switzerland
Xiao-Qiao He, China
Sebastian Heidenreich, Germany
Luca Heltai, Italy
Alfredo G. Hernández-Díaz, Spain
M.I. Herreros, Spain
Vincent Hilaire, France
Roland Hildebrand, Germany
Eckhard Hitzer, Japan
Jaromir Horacek, Czech Republic
Muneo Hori, Japan
András Horváth, Italy
Gordon Huang, Canada
Nicolas Hudon, Canada
Sajid Hussain, Canada
Asier Ibeas, Spain
Orest V. Iftime, Netherlands
Giacomo Innocenti, Italy
Emilio Insfran, Spain
Nazrul Islam, USA
Benoit Iung, France
Payman Jalali, Finland
Reza Jazar, Australia
Khalide Jbilou, France
Linni Jian, China
Bin Jiang, China
Zhongping Jiang, USA
Ningde Jin, China
Grand R. Joldes, Australia
Dylan F. Jones, UK
Tamas Kalmar-Nagy, Hungary
Tomasz Kapitaniak, Poland
Haranath Kar, India
Konstantinos Karamanos, Belgium
Jean-Pierre Kenne, Canada
Chaudry M. Khalique, South Africa
Do Wan Kim, Republic of Korea
Nam-Il Kim, Republic of Korea
Oleg Kirillov, Germany
Manfred Krafczyk, Germany
Frederic Kratz, France
Petr Krysl, USA
Jurgen Kurths, Germany
Kyandoghere Kyamakya, Austria
Davide La Torre, Italy
Risto Lahdelma, Finland
Hak-Keung Lam, UK
Jimmy Lauber, France
Antonino Laudani, Italy
Aime' Lay-Ekuakille, Italy
Nicolas J. Leconte, France
Marek Lefik, Poland
Yaguo Lei, China
Stefano Lenci, Italy
Roman Lewandowski, Poland
Panos Liatsis, UAE
Anatoly Lisnianski, Israel
Peide Liu, China
Peter Liu, Taiwan
Wanquan Liu, Australia
Yan-Jun Liu, China
Alessandro Lo Schiavo, Italy
Jean J. Loiseau, France
Paolo Lonetti, Italy
Sandro Longo, Italy
Sebastian López, Spain
Luis M. López-Ochoa, Spain
Vassilios C. Loukopoulos, Greece
Valentin Lychagin, Norway
Antonio Madeo, Italy
José María Maestre, Spain
Fazal M. Mahomed, South Africa
Noureddine Manamanni, France
Didier Maquin, France
Paolo Maria Mariano, Italy
Damijan Markovic, France
Benoit Marx, France
Franck Massa, France
Paolo Massioni, France
Ge'ard A. Maugin, France
Alessandro Mauro, Italy
Michael Mazilu, UK
Driss Mehdi, France
Roderick Melnik, Canada
Pasquale Memmolo, Italy
Xiangyu Meng, Canada
Jose Merodio, Spain
Alessio Merola, Italy
Luciano Mescia, Italy
Laurent Mevel, France
Yuri Vladimirovich Mikhlin, Ukraine
Aki Mikkola, Finland
Hiroyuki Mino, Japan
Pablo Mira, Spain
Vito Mocella, Italy
Roberto Montanini, Italy
Gisele Mophou, France
Rafael Morales, Spain
Marco Morandini, Italy
Simone Morganti, Italy

Philip Moser, Germany
Aziz Moukrim, France
Emiliano Mucchi, Italy
Domenico Mundo, Italy
Jose J. Muñoz, Spain
Giuseppe Muscolino, Italy
Marco Mussetta, Italy
Hakim Naceur, France
Hassane Naji, France
Keivan Navaie, UK
Dong Ngoduy, UK
Tatsushi Nishi, Japan
Xesús Nogueira, Spain
Ben T. Nohara, Japan
Mohammed Nouari, France
Mustapha Nourelfath, Canada
Sotiris K. Ntuyas, Greece
Roger Ohayon, France
Mitsuhiro Okayasu, Japan
Javier Ortega-García, Spain
Alejandro Ortega-Moñux, Spain
Naohisa Otsuka, Japan
Erika Ottaviano, Italy
Arturo Pagano, Italy
Alkis S. Paipetis, Greece
Alessandro Palmeri, UK
Anna Pandolfi, Italy
Elena Panteley, France
Achille Paolone, Italy
Xosé M. Pardo, Spain
Manuel Pastor, Spain
Pubudu N. Pathirana, Australia
Francesco Pellicano, Italy
Marcello Pellicciari, Italy
Haipeng Peng, China
Mingshu Peng, China
Zhike Peng, China
Marzio Pennisi, Italy
Matjaz Perc, Slovenia
Francesco Pesavento, Italy
Dario Piga, Italy
Antonina Pirrotta, Italy
Marco Pizzarelli, Italy
Vicent Pla, Spain
Javier Plaza, Spain
Sébastien Poncet, Canada
Jean-Christophe Ponsart, France
Mauro Pontani, Italy
Stanislav Potapenko, Canada
Sergio Preidikman, USA
Christopher Pretty, New Zealand
Carsten Proppe, Germany
Luca Pugi, Italy
Giuseppe Quaranta, Italy
Dane Quinn, USA
Vitomir Racic, Italy
Jose Ragot, France
Kumbakonam Rajagopal, USA
Gianluca Ranzi, Australia
Alain Rassineux, France
S.S. Ravindran, USA
Alessandro Reali, Italy
Oscar Reinoso, Spain
Nidhal Rezg, France
Ricardo Riaza, Spain
Gerasimos Rigatos, Greece
Eugenio Roanes-Lozano, Spain
Bruno G. M. Robert, France
José Rodellar, Spain
Rosana Rodriguez-Lopez, Spain
Ignacio Rojas, Spain
Carla Roque, Portugal
Debasish Roy, India
Gianluigi Rozza, Italy
Rubén Ruiz García, Spain
Antonio Ruiz-Cortes, Spain
Ivan D. Rukhlenko, Australia
Mazen Saad, France
Kishin Sadarangani, Spain
Mehrdad Saif, Canada
Miguel A. Salido, Spain
Roque J. Salterén, Spain
Francisco J. Salvador, Spain
Alessandro Salvini, Italy
Maura Sandri, Italy
Miguel A. F. Sanjuan, Spain
Juan F. San-Juan, Spain
Roberta Santoro, Italy
Ilmar Ferreira Santos, Denmark
José A. Sanz-Herrera, Spain
Nickolas S. Sapidis, Greece
Evangelos J. Sapountzakis, Greece
Andrey V. Savkin, Australia
Thomas Schuster, Germany
Mohammed Seaid, UK
Lotfi Senhadji, France
Joan Serra-Sagrsta, Spain
Gerardo Severino, Italy
Ruben Sevilla, UK
Leonid Shaikhet, Ukraine
Hassan M. Shanechi, USA
Bo Shen, Germany
Suzanne M. Shontz, USA
Babak Shotorban, USA
Zhan Shu, UK
Dan Simon, Greece
Luciano Simoni, Italy
Christos H. Skiadas, Greece
Michael Small, Australia
Alba Sofi, Italy
Francesco Soldovieri, Italy
Raffaele Solimene, Italy
Jussi Sopanen, Finland
Ruben Specogna, Italy
Sri Sridharan, USA
Ivanka Stamova, USA
Salvatore Strano, Italy
Yakov Strelniker, Israel
Sergey A. Suslov, Australia
Thomas Svensson, Sweden
Andrzej Swierniak, Poland
Yang Tang, Germany
Sergio Teggi, Italy
Alexander Timokha, Norway
Gisella Tomasini, Italy
Francesco Tornabene, Italy
Antonio Tornambe, Italy
Fernando Torres, Spain
Sébastien Tremblay, Canada
Irina N. Trendafilova, UK
George Tsiatas, Greece
Antonios Tsourdos, UK
Vladimir Turetsky, Israel
Mustafa Tutar, Spain
Ilhan Tuzcu, USA
Efstratios Tzirtzilakis, Greece
Filippo Ubertini, Italy
Francesco Ubertini, Italy
Hassan Ugail, UK
Giuseppe Vairo, Italy
Kuppalapalle Vajravelu, USA
Robertt A. Valente, Portugal
Pandian Vasant, Malaysia
Miguel E. Vázquez-Méndez, Spain



Josep Vehi, Spain
K. Veluvolu, Republic of Korea
Alessandro Veneziani, USA
Fons J. Verbeek, Netherlands
Franck J. Vernerey, USA
Georgios Veronis, USA
Anna Vila, Spain
Rafael Villanueva, Spain
Uchechukwu E. Vincent, UK
Mirko Viroli, Italy
Michael Vynnycky, Sweden

Shuming Wang, China
Yan-Wu Wang, China
Yongqi Wang, Germany
Roman Wendner, Austria
Desheng D. Wu, Canada
Yuqiang Wu, China
Guangming Xie, China
Xuejun Xie, China
Gen Qi Xu, China
Hang Xu, China
Joseph-Julien Yamé, France

Xing-Gang Yan, UK
Luis J. Yebra, Spain
Peng-Yeng Yin, Taiwan
Qin Yuming, China
Vittorio Zampoli, Italy
Ibrahim Zeid, USA
Huaguang Zhang, China
Qingling Zhang, China
Jian Guo Zhou, UK
Quanxin Zhu, China
Mustapha Zidi, France

Contents

Computational Intelligence in Image Processing 2016

Erik Cuevas, Daniel Zaldivar, Gonzalo Pajares, Marco Perez-Cisneros, and Raúl Rojas
Volume 2016, Article ID 5680246, 3 pages

Image-Guided Voronoi Aesthetic Patterns with an Uncertainty Algorithm Based on Cloud Model

Tao Wu and Limin Zhang
Volume 2016, Article ID 9837123, 18 pages

Learn to Rank Images: A Unified Probabilistic Hypergraph Model for Visual Search

Kaiman Zeng, Nansong Wu, Arman Sargolzaei, and Kang Yen
Volume 2016, Article ID 7916450, 7 pages

Color, Scale, and Rotation Independent Multiple License Plates Detection in Videos and Still Images

Narasimha Reddy Soora and Parag S. Deshpande
Volume 2016, Article ID 9306282, 14 pages

A Multikernel-Like Learning Algorithm Based on Data Probability Distribution

Guo Niu, Zhengming Ma, and Shuyu Liu
Volume 2016, Article ID 5927306, 18 pages

Hand Gesture Recognition Using Particle Swarm Movement

Clementine Nyirarugira, Hyo-rim Choi, and TaeYong Kim
Volume 2016, Article ID 1919824, 8 pages

An Automatic Cognitive Graph-Based Segmentation for Detection of Blood Vessels in Retinal Images

Rasha Al Shehhi, Prashanth Reddy Marpu, and Wei Lee Woon
Volume 2016, Article ID 7906165, 15 pages

Improved Adaptive Vibe and the Application for Segmentation of Complex Background

Le Chang, Zhenghua Liu, and Yan Ren
Volume 2016, Article ID 3835952, 8 pages

Contrast Enhancement Method Based on Gray and Its Distance Double-Weighting Histogram Equalization for 3D CT Images of PCBs

Lei Zeng, Bin Yan, and Weidong Wang
Volume 2016, Article ID 1529782, 13 pages

Neural Architectures for Correlated Noise Removal in Image Processing

Cătălina Cocianu and Alexandru Stan
Volume 2016, Article ID 6153749, 14 pages

Solving Adaptive Image Restoration Problems via a Modified Projection Algorithm

Hao Yang, Xueping Luo, and Leiting Chen
Volume 2016, Article ID 6132356, 11 pages

Hybrid Predictor and Field-Biased Context Pixel Selection Based on PPVO

Hongyin Xiang, Jinsha Yuan, and Sizu Hou
Volume 2016, Article ID 2585983, 16 pages



Normal Inverse Gaussian Model-Based Image Denoising in the NSCT Domain

Jian Jia, Yongxin Zhang, Li Chen, and Zhihua Zhao

Volume 2015, Article ID 851313, 13 pages

Single Image Superresolution Using Maximizing Self-Similarity Prior

Jianhong Li, Yarong Wu, Xiaonan Luo, Chengcai Leng, and Bo Li

Volume 2015, Article ID 312423, 10 pages

Editorial

Computational Intelligence in Image Processing 2016

Erik Cuevas,¹ Daniel Zaldívar,¹ Gonzalo Pajares,² Marco Perez-Cisneros,¹ and Raúl Rojas³

¹*Departamento de Electrónica, Universidad de Guadalajara, CUCEI, Avenida Revolución 1500, Guadalajara, JAL, Mexico*

²*Departamento de Ingeniería de Software e Inteligencia Artificial, Facultad Informática, Universidad Complutense, 28040 Madrid, Spain*

³*Institut für Informatik, Freie Universität Berlin, Arnimallee 7, 14195 Berlin, Germany*

Correspondence should be addressed to Erik Cuevas; erik.cuevas@cucei.udg.mx

Received 4 August 2016; Accepted 4 August 2016

Copyright © 2016 Erik Cuevas 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.

Computational intelligence (CI) has emerged as a powerful tool for information processing, decision-making, and knowledge management. CI approaches, in general, are useful for designing advanced computerized systems that possess useful characteristics mimicking human behaviors and capabilities in solving complex tasks, for example, learning, adaptation, and evolution. Examples of some popular CI models include fuzzy systems, artificial neural networks, evolutionary algorithms, multiagent systems, decision trees, rough set theory, knowledge-based systems, and hybrid of these models.

On the other hand, images have always played an essential role in human life. In the past they were, today they are, and in the future they will continue to be one of our most important information carriers. Recent advances in digital imaging and computer hardware technology have led to an explosion in the use of digital images in a variety of scientific and engineering applications. Therefore, each new approach that is developed by engineers, mathematicians, and computer scientists is quickly identified, understood, and assimilated in order to be applied to image processing problems.

Classical image processing methods often face great difficulties while dealing with images containing noise and distortions. Under such conditions, the use of computational intelligence approaches has been recently extended to address challenging real-world image processing problems. The interest on the subject among researchers and developers is increasing day by day as it is branded by huge volumes of research works that get published in leading international journals and international conference proceedings.

The main objective of this special issue is to bridge the gap between computational intelligence techniques and challenging image processing applications. Since this idea was first conceived, the goal has aimed at exposing the readers to the cutting-edge research and applications that are going on across the domain of image processing, particularly those whose contemporary computational intelligence techniques can be or have been successfully employed.

The special issue received several high-quality submissions from different countries all over the world. All submitted papers have followed the same standard of peer-reviewing by at least three independent reviewers, just as it is applied to regular submissions to the Mathematical Problems in Engineering journal. Due to the limited space, a very short number of papers have been finally included. The primary guideline has been to demonstrate the wide scope of computational intelligence algorithms and their applications to image processing problems.

The paper authored by T. Wu and L. Zhang presents an uncertainty algorithm based on cloud model for the generation of image-guided Voronoi aesthetic patterns. As a computational intelligence tool, cloud model handles the uncertainty more completely and more freely, and it cannot be considered as randomness compensated by fuzziness, fuzziness compensated by randomness, second-order fuzziness, or second-order randomness. To obtain the default parameters, authors conduct seven groups of experiments to test the proposed method. Using both visual and quantitative comparisons, T. Wu and L. Zhang prove the efficacy of the proposed method using two groups of experiments. Compared with the related methods, experimental results show

that the Voronoi-based aesthetic patterns with soft borders can be successfully generated by using the new technique.

K. Zeng et al. introduced a ranking model by understanding the complex relations within product visual and textual information in visual search systems. To understand their complex relations, authors focused on using graph-based paradigms to model the relations among product images, product category labels, and product names and descriptions. K. Zeng et al. developed a unified probabilistic hypergraph ranking algorithm, which, modeling the correlations among product visual features and textual features, extensively enriches the description of the image. The authors conducted experiments on the proposed ranking algorithm on a data set collected from a real e-commerce website. The results of their comparison demonstrate that the proposed algorithm extensively improves the retrieval performance over the visual distance based ranking.

N. R. Soora and P. S. Deshpande present a novel License Plate (LP) detection method using different clustering techniques, based on geometrical properties of the LP characters. In the paper, authors also propose a new character extraction method, for noisy/missed character components of the LP due to the presence of noise between LP characters and LP border. The proposed method detects the LP of any type of vehicle (including vans, cars, trucks, and motorcycles), having different plate variations, under different environmental and weather conditions because of the geometrical properties of the set of characters in the LP. The proposed method is independent of color, rotation, size, and scale variances of the LP. The concept is tested using standard media-lab and Application Oriented License Plate (AOLP) benchmark LP recognition databases. The success rate of the proposed approach for LP detection using media-lab database is 97.3% and using AOLP database is 93.7%. Results clearly indicate that the proposed approach is comparable to the previously published papers, which evaluated their performance on publicly available benchmark LP databases.

The paper authored by C. Nyirugira et al. presents a gesture recognition method derived from particle swarm movement for free-air hand gesture recognition. Under such conditions, authors suggest an automated process of segmenting meaningful gesture trajectories based on particle swarm movement. A subgesture detection and reasoning method is incorporated in the proposed recognizer to avoid premature gesture spotting. Evaluation of the proposed method shows promising recognition results: 97.6% on preisolated gestures, 94.9% on stream gestures with assistive boundary indicators, and 94.2% for blind gesture spotting on digit gesture vocabulary. The proposed recognizer requires fewer computation resources; thus it is a good candidate for real-time applications.

R. Al Shehhi et al. present a hierarchical graph-based segmentation for blood vessel detection in digital retinal images. This segmentation method employs some of perceptual Gestalt principles: similarity, closure, continuity, and proximity to merge segments into coherent connected vessel-like patterns. The integration of Gestalt principles is based on object-based features (e.g., color, black top-hat (BTH) morphology, and context) and graph-analysis algorithms

(e.g., Dijkstra path). The segmentation framework consists of two main steps: preprocessing and multiscale graph-based segmentation. Preprocessing is to enhance lighting condition, due to low illumination contrast, and to construct necessary features to enhance vessel structure due to sensitivity of vessel-patterns to multiscale/orientation structure. Graph-based segmentation is to decrease computational processing required for region of interest into most semantic objects. The segmentation was evaluated on three publicly available datasets. Experimental results show that preprocessing stage achieves better results compared to the state-of-the-art enhancement methods. The performance of the proposed graph-based segmentation is found to be consistent and comparable to other existing methods, with improved capability in detecting small/thin vessels.

The paper authored by G. Niu et al. proposes a multikernel-like learning algorithm based on data probability distribution (MKDPD) for classification proposes. In the approach, the parameters of a kernel function are locally adjusted according to the data probability distribution and thus produce different kernel functions. These different kernel functions will generate different Reproducing Kernel Hilbert Spaces (RKHS). The direct sum of the subspaces of these RKHS constitutes the solution space of the learning problem. Furthermore, based on the proposed MKDPD algorithm, an algorithm for labeling new coming data is also introduced, in which the basic functions are retrained according to the new coming data, while the coefficients of the retrained basic functions remained unchanged to label the new coming data. The experimental results presented in this paper show the effectiveness of the proposed algorithms.

H. Yang et al. introduce a new general TV regularizer, namely, generalized TV regularization, to study image denoising and nonblind image deblurring problems. In order to discuss the generalized TV image restoration with solution-driven adaptivity, authors consider the existence and uniqueness of the solution for mixed quasivariational inequality. Moreover, the convergence of a modified projection algorithm for solving mixed quasivariational inequalities is also shown. The corresponding experimental results support our theoretical findings.

C.-L. Cocianu and A. Stan propose a new method that combines the decorrelation and shrinkage techniques to neural network-based approaches for noise removal purposes. The images are represented as sequences of equal sized blocks, each block being distorted by a stationary statistical correlated noise. Some significant amount of the induced noise in the blocks is removed in a preprocessing step, using a decorrelation method combined with a standard shrinkage-based technique. The preprocessing step provides for each initial image a sequence of blocks that are further compressed at a certain rate, each component of the resulted sequence being supplied as inputs to a feed-forward neural architecture. The local memories of the neurons of the layers are generated through a supervised learning process based on the compressed versions of blocks of the same index value supplied as inputs and the compressed versions of them resulted as the mean of their preprocessed versions. Finally, using the standard decompression technique, the sequence of the

decompressed blocks is the cleaned representation of the initial image. The performance of the proposed method is evaluated by a long series of tests, the results being very encouraging as compared to similar developments for noise removal purposes.

The paper authored by L. Chang et al. introduces a method to solve the problems which basic Vibe algorithm cannot effectively eliminate such as the influence of background noise, follower shadow, and ghost under complex background. Therefore, considering the basic Vibe algorithm, this paper puts forward some improvement measures in threshold setting, shadow eliminating, and ghost suppression. Firstly, judgment threshold takes adjustment with the changes of background. Secondly, a fast eliminating ghost algorithm depending on adaptive threshold is introduced. Finally, follower shadow is detected and inhibited effectively through the gray properties and texture characteristics. Experiments show that the proposed algorithm works well in complex environment without affecting computing speed and has stronger robustness and better adaptability than the basic algorithm. Meanwhile, the ghost and follower shadow can be absorbed quickly as well. Therefore, the accuracy of target detection is effectively improved.

L. Zeng et al. propose an image enhancement algorithm to solve the well-known problems that involve the detection methods for 3D nondestructive testing of printed circuit boards (PCBs). Therefore, considering the characteristics of 3D CT images of PCBs, the proposed algorithm uses gray and its distance double-weighting strategy to change the form of the original image histogram distribution, suppresses the grayscale of a nonmetallic substrate, and expands the grayscale of wires and other metals. The proposed algorithm also enhances the gray difference between a substrate and a metal and highlights metallic materials. The proposed algorithm can enhance the gray value of wires and other metals in 3D CT images of PCBs. It applies enhancement strategies of changing gray and its distance double-weighting mechanism to adapt to this particular purpose. The flexibility and advantages of the proposed algorithm are confirmed by analyses and experimental results.

The paper authored by H. Xiang et al. presents a pixel-value-ordering hybrid algorithm for error prediction in images. The proposed method predicts pixel in both positive and negative orientation. Assisted by expansion bins selection technique, this hybrid predictor presents an optimized prediction-error expansion strategy including bin 0. Furthermore, a novel field-biased context pixel selection is already developed, with which detailed correlations of around pixels are better exploited more than equalizing scheme merely. Experiment results show that the proposed approach improves embedding capacity and enhances marked image fidelity. It also outperforms some other state-of-the-art methods of reversible data hiding, especially for moderate and large payloads.

J. Jia et al. introduce a novel normal inverse Gaussian model-based method that uses a Bayesian estimator to carry out image denoising in the nonsubsampling contourlet transform (NSCT) domain. In the proposed method, the model is first used to describe the distributions of the image transform

coefficients of each subband in the NSCT domain. Then, the corresponding threshold function is derived from the model using Bayesian maximum a posteriori probability estimation theory. Finally, optimal linear interpolation thresholding algorithm (OLI-Shrink) is employed to guarantee a gentler thresholding effect. The results of comparative experiments conducted indicate that the denoising performance of our proposed method in terms of peak signal-to-noise ratio is superior to that of several state-of-the-art methods, including BLS-GSM, K-SVD, BivShrink, and BM3D. Further, the proposed method achieves structural similarity (SSIM) index values that are comparable to those of the block-matching 3D transformation (BM3D) method.

The paper authored by B. Li et al. develops a new approach for solving the problem of single image superresolution by generalizing this property. The main idea of this approach takes advantage of a generic prior that assumes a randomly selected patch in the underlying high resolution (HR) image should visually resemble as much as possible with some patch extracted from the input low resolution (LR) image. Under such conditions, this approach deploys a cost function and applies an iterative scheme to estimate the optimal HR image. For solving the cost function, authors introduce Gaussian mixture model (GMM) to train upon a sampled data set for approximating the joint probability density function (PDF) of input image with different scales. Through extensive comparative experiments, this paper demonstrates that the visual fidelity of our proposed method is often superior to those generated by other state-of-the-art algorithms as determined through both perceptual judgment and quantitative measures.

Acknowledgments

Finally, we would like to express our gratitude to all of the authors for their contributions and the reviewers for their efforts to provide valuable comments and feedback. We hope this special issue offers a comprehensive and timely view of the area of applications of computational intelligence in image processing and that it will grant stimulation for further research.

*Erik Cuevas
Daniel Zaldívar
Gonzalo Pajares
Marco Perez-Cisneros
Raúl Rojas*

Research Article

Image-Guided Voronoi Aesthetic Patterns with an Uncertainty Algorithm Based on Cloud Model

Tao Wu^{1,2} and Limin Zhang²

¹*School of Information Science and Technology, Lingnan Normal University, Zhanjiang 524048, China*

²*Guangdong Engineering and Technological Development Center for E-Learning, Zhanjiang 524048, China*

Correspondence should be addressed to Tao Wu; wutao@whu.edu.cn

Received 3 March 2016; Revised 17 June 2016; Accepted 26 June 2016

Academic Editor: Marco Perez-Cisneros

Copyright © 2016 T. Wu and L. Zhang. 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.

Tessellation-based art is an important technique for computer aided aesthetic patterns generation, and Voronoi diagram plays a key role in the preprocessing, whose uncertainty mechanism is still a challenge. However, the existing techniques handle the uncertainty incompletely and unevenly, and the corresponding algorithms are not of high efficiency; thus it is impossible for users to obtain the results in real time. For a reference image, a Voronoi aesthetic pattern generation algorithm with uncertainty based on cloud model is proposed, including uncertain line representation using an extended cloud model and Voronoi polygon approximation filling with uncertainty. In view of the different parameters, seven groups of experiments and various experimental analyses are conducted. Compared with the related algorithms, the proposed technique performs better on running time, and its time complexity is approximatively linear related to the size of the input image. The experimental results show that it can produce visually similar effect with the frayed or cracked soil and has three advantages, that is, uncertainty, simplicity, and efficiency. The proposal can be a powerful alternative to the traditional methods and has a prospect of applications in the digital entertainment, home decoration, clothing design, and various fields.

1. Introduction

Computational aesthetics is a major unsolved problem in computer science and engineering [1, 2]. Over the last decade, there has been increasing interest in using computational intelligence approaches to solve this problem [3–8]. Of those, tessellation-based Voronoi art is an important technique and can be widely applied in various fields, that is, architecture, jewellery design, fashion design, and so forth, and Voronoi diagram plays a key role in its preprocessing [9]. In fact, tessellation-based or area-based visual representations are common to many artistic works or computer-based visualization systems. Also, it would be beneficial to researchers to develop and test the computational intelligence algorithms for creating interesting and aesthetic images and videos based on Voronoi diagram.

In recent years, several approaches to create nicely looking patterns have been described in literature. For example, Kaplan first proposed Voronoi-based art and applied it to

decorative design [10]; the similar style includes portrait stylization [11], image filter [12], and inscribed curve [13]. These above methods fall into Voronoi generation with an accurate perspective. More recently, some uncertain methods with computational intelligence algorithms have been surfaced and received some attentions, such as the fuzzy border of Voronoi polygon [14] and the probabilistic Voronoi model [15, 16]. In addition, the uncertain Voronoi structure also exists in other techniques for computational aesthetics, and Isenberg [17] and Kim et al. [7] proposed the abstraction method, respectively. Based on this, Michael et al. provided a random method for Voronoi art [8].

Nonetheless, each method has its own advantages, disadvantages, and applied situations. The existing methods with uncertainty are based on fuzzy set or probability statistics, and there are some drawbacks of the processing, and their results are unsatisfactory or even questionable in some cases. Firstly, the existing methods cannot completely capture the uncertainty in the Voronoi generation. The fuzzy Voronoi

diagram theoretically extends the border as a transition region, which is still a continuous plane, not applicable to the image stylization, while the random Voronoi diagram only adds a shadow to the real boundary. In fact, the uncertainty includes fuzziness, randomness, and the connection of them. Thus, any partial solution to only one of these aspects that does not take the other into account is incomplete. Secondly, the existing methods with uncertainty are not very efficient. The rendering is point-based operation, which is simple but time-consuming and difficult to apply in real-time production. We believe that the mathematical representation of a concept with uncertainty is one of the foundations of computational intelligence, and uncertainty is an inherent part of Voronoi art in real world applications; then the aesthetic generation with uncertainty is still a challenge. Therefore, we should further extend and discuss the traditional approaches from a developmental point of view. Cloud model, different from statistical methods and fuzzy set methods, can handle such uncertainty in a better way, since it provides us with more design degrees of freedom, at least the second-order uncertainty.

In this context, we proposed an image-guided Voronoi aesthetic patterns with an uncertainty algorithm based on cloud model (iVPC for short). It is noted that what it meant is not the Virtual Private Cloud, and the cloud model is completely different from cloud computing. Our intentions are threefold: (1) how can the image-guided Voronoi art be produced using a cloud model-based algorithm? (2) What is a suitable configuration of the cloud model-based algorithm for image-guided Voronoi art? (3) How different types of Voronoi decomposition can affect the aesthetic qualities of the rendered images?

Cloud model, compared to similar techniques, fuzzy set and rough set, is a cognitive model between a qualitative concept and its quantitative instantiations and successfully used in various applications [18–22]. The cloud model-based rendering consists of four main steps, including random reference points-based or image reference-based Voronoi decomposition, uncertain line representation using the extended cloud model, uncertain Voronoi polygon approximation filling, and then generating an aesthetic image with the Voronoi art style.

Our method satisfies the USE property, and it stands for uncertainty, simplicity, and efficiency: (1) the proposed method is involved as a novel technique with uncertainty using cloud model, although there have been some efforts in using computational intelligence algorithms. (2) The proposed method is very simple, and only the uncertain representation step on Voronoi polygon and its border are introduced into the classical Voronoi-based method. (3) From the perspective of running time, the proposed method is efficient, and its time complexity is approximately linear with the size of the original image for image-guided decomposition.

The remainder of this paper is organized as follows. We describe our iVPC algorithm in detail: first uncertain line using cloud model in Section 2, next uncertain Voronoi polygon in Section 3, and then cloud model-based algorithm and its time complexity in Section 4. In Section 5, we

investigate the parameter configuration and show several examples of the resulting images and then conduct four groups of experiments, including visual comparisons, time performance analysis, quantitative comparisons, and the user study. Finally, we discuss the results and give some ideas for future improvements in Section 6.

2. Uncertain Line Using Cloud Model

2.1. Cloud Model. Cloud model, proposed by Li et al. [18, 19], is the innovation and development of membership function in fuzzy theory and uses probability and mathematical statistics to analyse the uncertainty [19]. In theory, there are several forms of cloud model, successfully used in various applications, including image processing, data mining, geological analysis, and knowledge engineering [18–22]. However, the normal cloud model is commonly used in practice, and its theoretical foundation is the universality of normal distribution and bell membership function [18, 21].

Let U be a universe set described by precise numbers, and let C be a qualitative concept related to U . Given a number $u \in U$, which randomly realizes the concept C , u satisfies $u \sim N(\text{Ex}, \text{En}n^2)$, where $\text{En}n \sim N(\text{En}, \text{He}^2)$, and the certainty degree of u on U is as below:

$$\mu(u) = \exp\left(-\frac{(u - \text{Ex})^2}{2\text{En}n^2}\right), \quad (1)$$

and then the distribution of u on U is defined as a normal cloud, and u is as a cloud drop.

The overall property of a concept $C(\text{Ex}, \text{En}, \text{He})$ can be represented by three numerical characters of normal cloud model, expected value Ex , entropy En , and hyper-entropy He . Ex is the mathematical expectation of the cloud drop distributed in universal set. En is the uncertainty measurement of the qualitative concept, and it is determined by both randomness and fuzziness of the concept. He is the uncertain measurement of entropy, which is determined by randomness and fuzziness of entropy En [18]. It is worth noting that hyper-entropy He of a cloud model is a deviation measure from a normal distribution; hence, the distribution of cloud drops can be regarded as a generalized normal distribution.

The kernel of normal cloud model is the transform between qualitative concept and quantitative data [21], and it is realized by normal cloud generators. On one hand, forward normal cloud generator is the mapping from qualitative concept to quantitative values, and it produces the cloud drops to describe a concept when three numerical characters and the number of cloud drops are input. On the other hand, backward normal cloud generator provides the transformation from quantitative numerical values to quality concept, and a normal cloud model with three numerical characters is defined by computing mean, absolute central moment with the first order, and variance of sample data. Essentially, the normal cloud generators are two algorithms based on probability measure space [18]; these processes are uncertain and cannot be expressed by a precise function. More information about normal cloud model can be obtained from [18].

2.2. *Uncertain Line.* According to the above idea, a line segment can also be represented by an extended cloud model, if considering it as a qualitative concept with the numerical characters. Arbitrary line is a part of a continuous domain in 2D plane, and an uncertain transformation from the quantitative value and the qualitative concept can be easily achieved using a cloud model. Based on the traditional Gaussian cloud model, we define the uncertain line and set of sampling points.

Given two endpoints of a line segment, that is, $\mathbf{P}_1 = (px_1, py_1)$ and $\mathbf{P}_2 = (px_2, py_2)$, where px_1, py_1, px_2, py_2 are the coordinates, we can take the slope k and the vertical intercept b as a key concept to be expressed by cloud model, which is determined by (Ex_k, En_k, He_k) and (Ex_b, En_b, He_b) . Consequently, a line segment would be defined by the parameter set $\{k, b\}$ as below:

$$\left(\begin{bmatrix} Ex_k \\ Ex_b \end{bmatrix}, \begin{bmatrix} En_k \\ En_b \end{bmatrix}, \begin{bmatrix} He_k \\ He_b \end{bmatrix} \right), \quad (2)$$

where Ex_k, Ex_b are the real values of slope and vertical intercept can be calculated by the following:

$$\begin{aligned} Ex_k &= \frac{py_2 - py_1}{px_2 - px_1}, \\ Ex_b &= py_1 - px_1 \frac{py_2 - py_1}{px_2 - px_1}. \end{aligned} \quad (3)$$

Once a vertical line is involved into the above equation, that is, $px_2 = px_1$, the slope does not exist, the abscissa should be swapped with the ordinate, and the only thing is an *if* statement using matrix transposition in program implementation. Even so, the abscissa values of two endpoints in Voronoi polygons are almost impossibly the same.

Let D , consisting of quantitative values, be a continuous domain in the plane, and let a line L be the qualitative concept related to D . For a line $l \in L$, it randomly realizes the concept L , which is determined by the slope k and the vertical intercept b . In addition, k, b satisfy $k \sim N(Ex_k, Enn_k^2)$ and $b \sim N(Ex_b, Enn_b^2)$, where $Enn_k \sim N(En_k, He_k^2)$ and $Enn_b \sim N(En_b, He_b^2)$, and the certainty degree of l on L is as below:

$$\mu = \mu_k \mu_b, \quad \mu_k = e^{-\frac{(k-Ex_k)^2}{2Enn_k^2}}, \quad \mu_b = e^{-\frac{(b-Ex_b)^2}{2Enn_b^2}}. \quad (4)$$

Each random realization l of a concept L is expressed by a set of points; then the distribution of l on D is defined as an extended Gaussian cloud model, and μ is as the certainty degree. For simplicity, we only take one point instead of a set of points. (lx_i, ly_i) ($i = 1, 2, \dots, n$) denotes the selected point on the i th random realization, corresponding to the line l_i , and n is the number of random process. The point set $\{(lx_1, ly_1), (lx_2, ly_2), \dots, (lx_n, ly_n)\}$, corresponding to all the cloud drops, constitutes a cloud concept on the line, and the approximate entity can be called uncertain line.

Different cloud drops, with various parameters and from a Gaussian cloud model, correspond to different slope and intercept, and all of these cloud drops are one of forms on

the line segment in 2D space. A given line segment can be approximated by n discrete cloud drops using a generation algorithm, and then the quantitative value and the qualitative concept can be transformed to each other uncertainly. The detailed algorithm is as follows.

Step 1. Given \mathbf{P}_1 and \mathbf{P}_2 , swap the abscissa with the ordinate if $px_2 = px_1$.

Step 2. According to (3), calculate the expected values $[Ex_k, Ex_b]^T$, and set the entropy $[En_k, En_b]^T$ and hyper-entropy $[He_k, He_b]^T$.

Step 3. Generate a normal random number Enn_k using the expectation En_k and the variance He_k^2 and another random number k_i using the expectation Ex_k and the variance Enn_k^2 .

Step 4. Generate a normal random number Enn_b using the expectation En_b and the variance He_b^2 and another random number b_i using the expectation Ex_b and the variance Enn_b^2 .

Step 5. For the selected point on the i th random realization l_i , obtain the abscissa value $lx_i = \text{rand}(1)(px_2 - px_1) + px_1$, where $\text{rand}(1)$ denotes a random number in $[0, 1]$, and then calculate the ordinate value $ly_i = k_i lx_i + b_i$.

Step 6. Calculate the certainty degree μ_i using (4) and then take this line l_i with certainty degree μ_i as a cloud drop on the concept L , and (lx_i, ly_i) is one of the discrete random samples. Of course, the number of cloud drops increases by 1.

Step 7. Repeat Steps 3 to 6 until the number of cloud drops reaches the predefined value n .

Given the endpoints $\mathbf{P}_1 = (0, 1)$ and $\mathbf{P}_2 = (4, 9)$, the expected values are calculated as $Ex_k = 2$ and $Ex_b = 1$, and the other numerical characters are fixed as $En_k = 0.1 Ex_k = 10He_k$ and $En_b = 0.1 Ex_b = 10He_b$. We can generate 50 cloud drops, and the location relations between the sample points and the line are drawn on the coordinate system. As shown in Figure 1(a), these sample points are around the line segment with a nonuniform distribution. Intuitively, most of offset values between the sample points and the line segment are acceptable, and then a concept "near or around the line segment" is achieved. In this point of view, an uncertain representation of a line segment is feasible and reasonable using the extended cloud model.

Next, we generate 500 cloud drops, and the relations among μ_k, μ_b , and μ are listed in Figure 1(b). With the increase of μ_k and μ_b , μ is also increased. The distribution of cloud drops near the peak is more intensive, and these cloud drops contribute more to representing the concept. It is noted that the certainty degree is a fuzzy measure on the concept, whose calculation itself is a fuzzy process (as Step 6), and this fuzzy measure characterizes the membership degree of sampling and reflects the value ranges of the accepted sample points on the domain space. But other than the pure fuzzy problems, randomness is also involved, since the sample points depend on the abscissa values of random sampling, as Step 5,

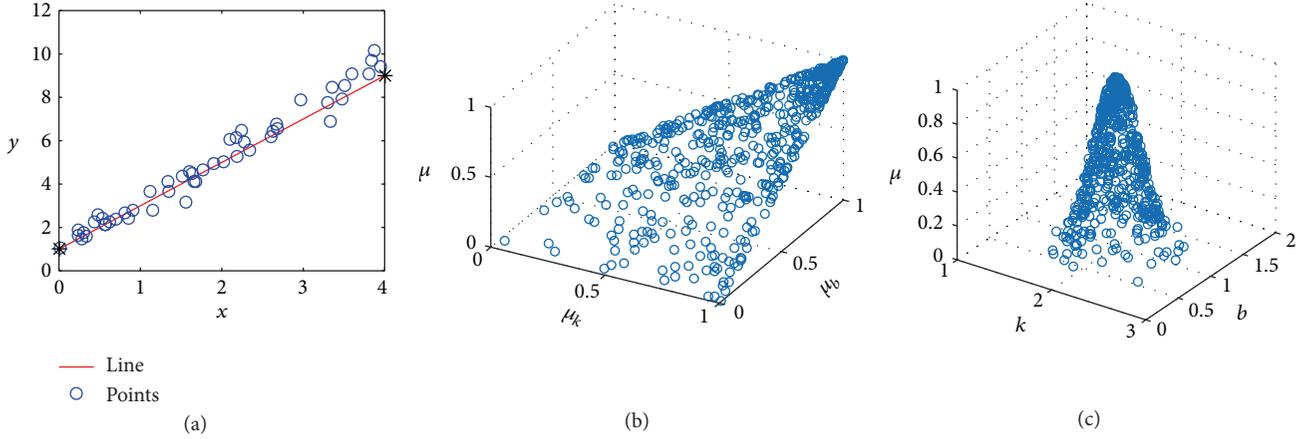


FIGURE 1: Uncertain line and its cloud drops: (a) a line with the sample points, (b) the joint distribution of certainty degree for μ_k, μ_b , and (c) the joint distribution of certainty degree k, b .

the random measure indicates the dispersion of cloud drops representing a line concept.

Furthermore, the relations among k , b , and μ are listed in Figure 1(c). Different groups of cloud drops play the different roles to the same concept, and there is a thick and dense distribution around the peak. With the slope k and the intercept b closing to the expected values Ex_k, Ex_b , the certainty degree of cloud drop is also larger, and it is coincident with the result in Figure 1(b). In fact, $\mu_k \rightarrow 1, \mu_b \rightarrow 1$ with $k \rightarrow Ex_k, b \rightarrow Ex_b$, and there is a close correlation between these two according to (4). The determination on the slope and the intercept is a random process, characterizing the dispersion degree of the line sampling, but it is not just a pure random problem; the associated certainty degree reflects the fuzziness.

In general, cloud model studies the randomness of membership grade, which is considered as fuzziness, randomness, and the connection of them. The proposed processing of randomness and fuzziness is not independent, and then it would take into account the correlation of them. The probabilistic Voronoi methods used expectation, variance, and other statistical characteristics, which reflected the randomness, but did not touch the fuzziness. While the fuzzy Voronoi theory calculated the membership degree using an accurate method, it failed to capture the randomness. We represent an uncertain line using cloud model and take into account the fuzziness, the randomness, and the relationship in between. Cloud model-based method can describe the concept “near a line segment,” which is more in accord with human cognition.

There is not a simple line processing in the Voronoi boundaries, and then an intuitive idea is to process and draw each boundary sequentially. Given a Voronoi polygon with four vertices, $\mathbf{P}_1 = (0, 1)$, $\mathbf{P}_2 = (4, 9)$, $\mathbf{P}_3 = (2, -15)$, and $\mathbf{P}_4 = (-1, -5)$, the uncertain expression is shown in Figure 2, and the sample points approximate the four lines.

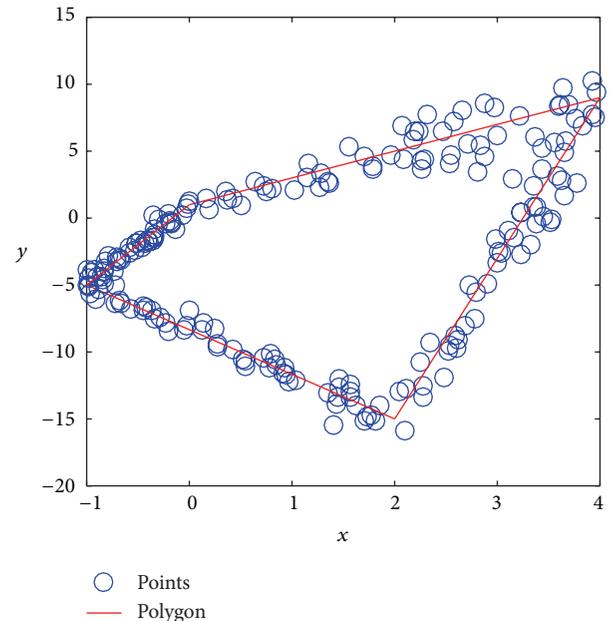


FIGURE 2: Uncertain polygon and its points.

3. Uncertain Voronoi Polygon Using Cloud Model

3.1. Image-Guided Voronoi Decomposition. For image-guided Voronoi art, the generation of Voronoi diagram is completely dependent on the image content. There exist various techniques, including image segmentation [23] and half-tone processing [24]. In the following, we use the Floyd-Steinberg dithering algorithm [25] since it is a simple, popular, and efficient method based on error diffusion. If involving the user interaction, Photoshop and other popular software can also be used to obtain a half-tone image.

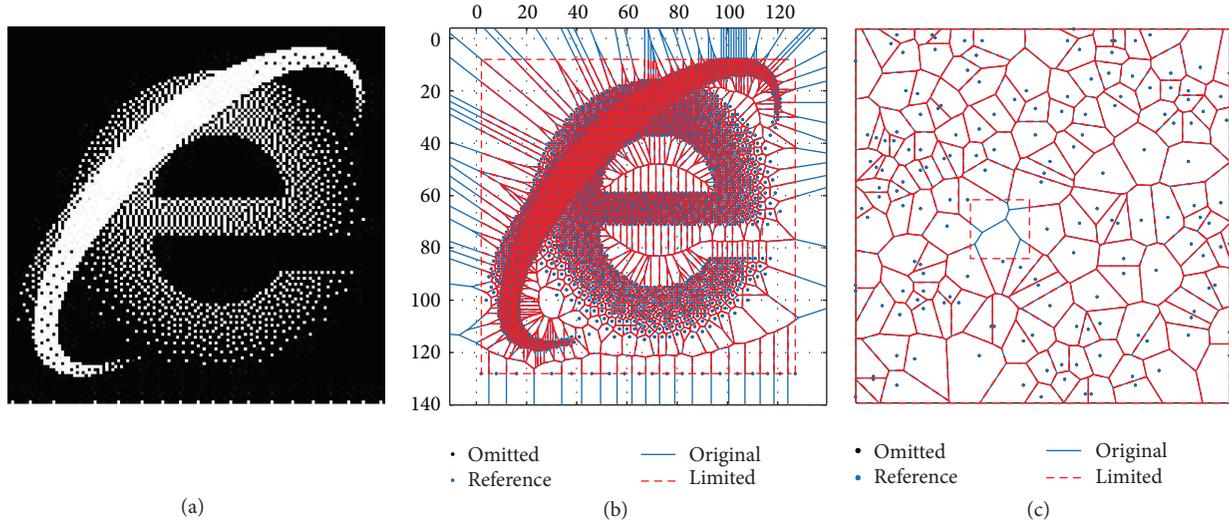


FIGURE 3: Voronoi decomposition: (a) the dithered IE logo image, (b) Voronoi decomposition for (a), and (c) an unsupervised decomposition.

As shown in Figure 3(a), the IE logo is processed by Floyd-Steinberg dithering algorithm. The pixels with nonzero grayscale values are selected and their coordinate values are as the reference points to generate Voronoi polygons. The number of the reference points is denoted by N , which is automatically determined by the Floyd-Steinberg algorithm. Then the Voronoi decomposition is generated with an outer-boundary constraint, and we use the VoronoiLimit method proposed by Jakob [26]. The decomposition result is shown in Figure 3(b), where the dot represents a reference point, and the dotted line represents a new constrained boundary.

We call the above decomposition supervised decomposition, which is image-guided. Moreover, the reference point can be also produced by a random method, that is, unsupervised decomposition.

Given the width w , the height h , and the density factor ρ , the reference points are randomly generated, and the Voronoi decomposition result is varied for each different running. The number of the reference points is $N = \text{round}(hw\rho)$.

Figure 3(c) shows an example of unsupervised decomposition; there is a rectangular area containing an internal hole. Also for simplicity, we investigate the parameters with the unsupervised decomposition in Section 5.

3.2. Uncertain Voronoi Aesthetic Patterns. After the Voronoi polygons are determined by Voronoi decomposition, the rendering can be done in a natural way. A line from each reference point to each sample point is drawn and filled one by one, and all lines in each polygon are drawn with the same color, while different polygons are rendered with various colors. Then the aesthetic patterns with different styles can be produced.

Taking the quadrilateral in Figure 2 as an example, the reference point is the centroid of the polygon $\mathbf{P}_r = (5/4, -5/2)$; the filling result is shown in Figure 4(a). The lines mainly cover the triangular region determined by the endpoints of the line and the reference point. But the distribution of lines

is not uniform, and some are sparse and some are intensive. In other words, the number of cloud drops may affect the aesthetic patterns.

In the case of enough number of cloud drops, we add an extra parameter related with the certainty degree, that is, intensity factor τ , and the aesthetic pattern can be controlled by removing the cloud drops and their sample points, whose certainty degree is less than τ .

Similarly, all uncertain lines are processed one by one, and the polygons are rendered with various styles. Although the method is very simple, there are two problems not solved: (1) near the vertices of polygons, the sample points from the adjacent borders are different; then the filling would cause the cross or overlap of lines. (2) The iterative rendering is of low efficiency and is time-consuming. To avoid the weaknesses and improve the algorithm performance, we provide the following method to process the uncertain polygon, which is an improved version of the uncertain line.

Given the t -sided polygon of a Voronoi decomposition, the vertex set $\{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_t\}$ ($t \geq 3$), and its centroid $\mathbf{P}_r = (px_r, py_r)$ (also as the reference point), the sides are determined by $\mathbf{P}_1\mathbf{P}_2, \mathbf{P}_2\mathbf{P}_3, \dots, \mathbf{P}_{t-1}\mathbf{P}_t, \mathbf{P}_t\mathbf{P}_1$, and each side is represented by n cloud drops according to the previous section; then the polygon corresponds to nt sample points. Consequently, all the cloud drops are determined by a point set $\{(lx_1, ly_1), (lx_2, ly_2), \dots, (lx_{nt}, ly_{nt})\}$, and their polar angles can be calculated as below:

$$\theta_j = \arg \tan (py_r - ly_j, px_r - lx_j), \quad (5)$$

where $j = 1, 2, \dots, nt$.

Once the polar angles are obtained, all sampling points are ordered and reorganized clockwise, taking the reference point as the center, and the duplicate samples are removed; then a polygon is simultaneously filled by the fill() function in Matlab.

Following this step, the aforementioned quadrilateral is filled, and 200 cloud drops are generated. The result is shown

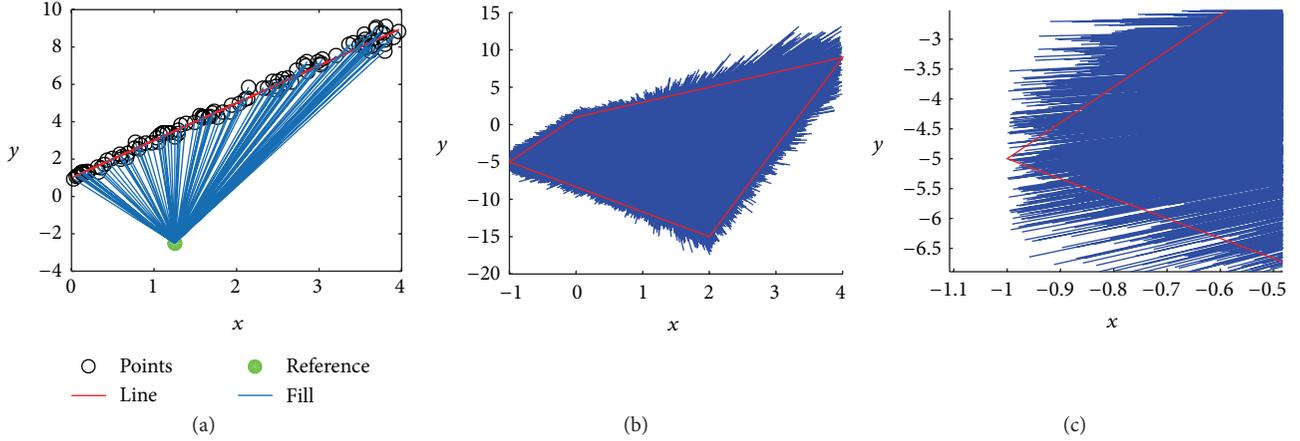


FIGURE 4: Filling uncertain polygon: (a) lines from the reference points and the sample points, (b) the filled polygon, and (c) the detail with an enlarged scale for (b).

in Figure 4(b); there are rough boundaries instead of the original accurate ones. As shown in Figure 4(c), there is the detail with an enlarged scale near P_4 . The number of sampling points is enough, but the fill result is delicate, without halting and with no overlapping.

4. The Proposed Algorithm

4.1. The Overview of iVPC Algorithm. To sum up, the proposed iVPC algorithm includes five main steps, and it is described as the following.

Step 1 (initialization). The parameters include the type of Voronoi decomposition, the numerical characteristics of cloud model $[En_k, En_b]^T$ and $[He_k, He_b]^T$, the number of cloud drops n , the number of the reference points which is denoted by N , the intensity factor τ , the colormap, and the selected background color. It should be noted that N is determined by the width w , the height h , and the density factor ρ for unsupervised decomposition, while, for supervised decomposition, N is autoset by the Floyd-Steinberg algorithm.

Step 2 (Voronoi decomposition). For image-guided decomposition, the reference points are produced by Floyd-Steinberg dithering algorithm. For unsupervised decomposition, the reference points are produced randomly. Then N Voronoi polygons are obtained.

Step 3 (uncertain line). For the c th Voronoi polygon, the cloud drops and sample points for t sides are generated by the method in Section 2.

Step 4 (uncertain Voronoi polygon). Voronoi polygon is approximated and filled using the method in Section 3.

Step 5 (loop). The counter c is incremented by $c = c + 1$. Repeat Steps 3 and 4 until all the Voronoi polygons are processed; that is, $c = N$.

4.2. Time Complexity. Step 1 is a simple initialization; the time complexity is $O(1)$, and it is not the key step when analysing the time performance. Step 2 is the necessary time cost for all of algorithms on Voronoi art generation, which depends on the decomposition technique. Theoretically, this step can be as a preprocessing. In Step 3, n cloud drops consume the time of order $O(n)$, and for a t -sides polygon, the iteration takes time $O(nt)$. In Step 4, the time cost in a single loop is mainly the sort of polar angle; the average time is $O((nt) \log(nt))$, and the other operations take time $O(nt)$ in the worst case, and $O(1)$ in the best case. In addition, Steps 3 and 4 are repeated N times. Thus, the time complexity of these two steps would approximately be $O(Nnt)$. Although t is different for all the Voronoi polygons, it is not very large, generally less than 6. That is to say, $t \ll Nn$.

In general, our algorithm costs time $O(Nn)$. More accurately, it is $O(hwpn)$ in the case of unsupervised decomposition, and $\rho n \ll hw$ is necessary for aesthetic reason; we will further investigate this point. A similar condition is also necessary for image-guided decomposition; that is, N cannot be too large. Therefore, the time complexity of the proposed algorithm is approximately linear with the size of the original image for image-guided decomposition, and the time complexity analysis indicates that our method is of high efficiency.

5. Experimental Results

5.1. Parameter Configuration. In order to investigate the parameter configuration of the proposed algorithm, both image-guided decomposition and unsupervised decomposition are considered, and we conduct seven groups of experiments.

5.1.1. Group 1. In this group, we test the density factor ρ for unsupervised decomposition. Given $h = w = 128$, three aesthetic patterns are generated using $\rho = 0.001, 0.01, 0.05$, respectively. As shown in Figure 5(a), it is indicated that the lower the density factor, the larger the area of the Voronoi

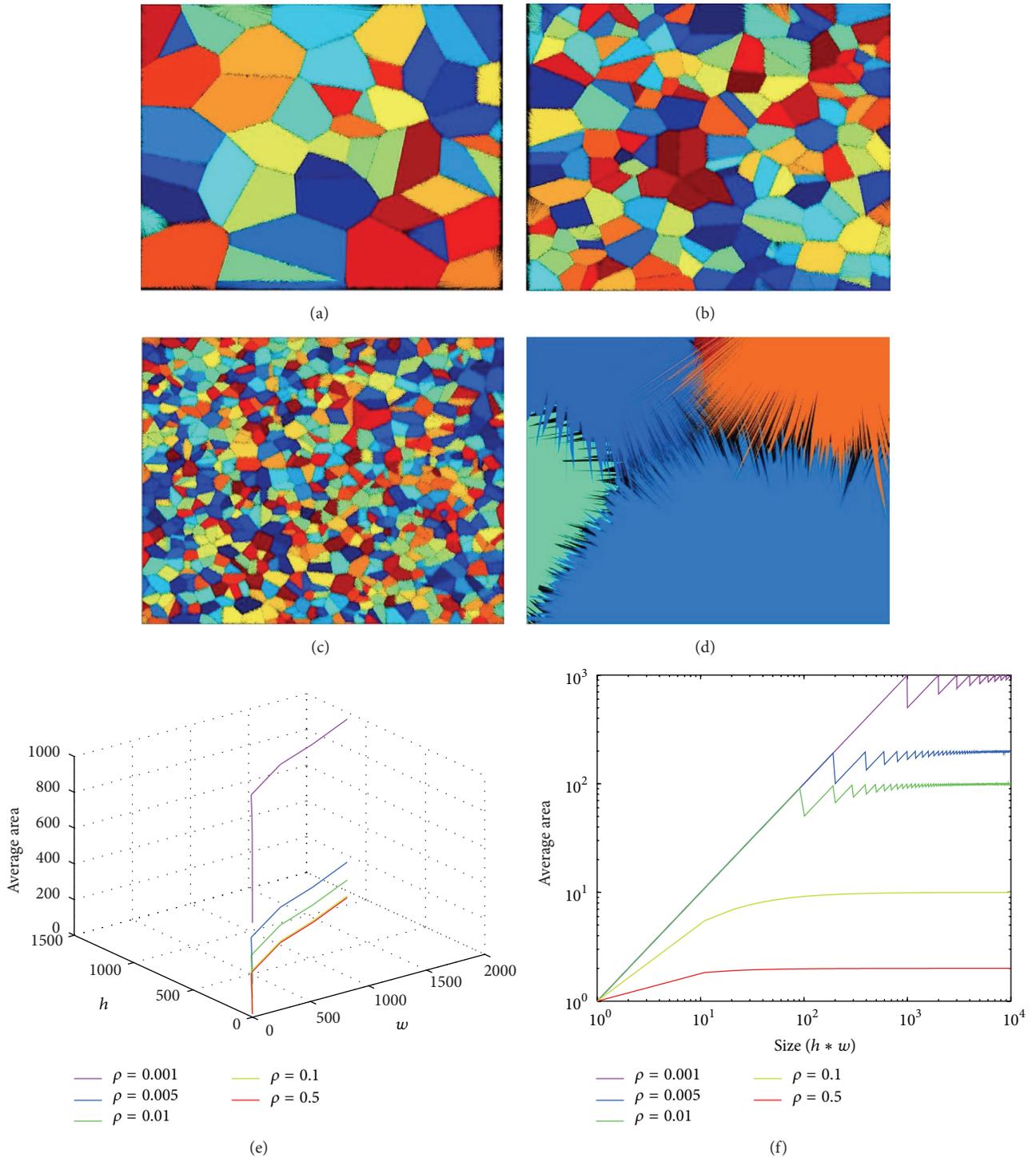


FIGURE 5: Voronoi decomposition with various densities: (a) $\rho = 0.001$, (b) $\rho = 0.01$, (c) $\rho = 0.05$, (d) the detail with an enlarged scale for (b), (e) average area of each Voronoi polygon varied with the height and the width, and (f) average area of each Voronoi polygon varied with the image size.

polygon and the less aesthetics the result pattern. While when the density factor is too high, the area of Voronoi polygon should be too small and an inordinate number of the reference points and extremely time-consuming performance would be inevitable, an example is shown in Figure 5(c).

Therefore, a moderate density factor is beneficial to the aesthetic effect.

The whole image is as Figure 5(b), which is with a proper density of reference points, a reasonable number of polygons, and better aesthetics. For detailed analysis, a subimage from

TABLE 1: Statistical results on cloud drops and their contribution with various cloud models.

| Cloud models | | | Mean and standard deviation | | | | | | | |
|--------------|-----|------|-----------------------------|-------|--------------------|-------|----------------------|-------|----------------------|-------|
| Ex | En | He | [Ex - 0.67En, Ex + 0.67En] | | [Ex - En, Ex + En] | | [Ex - 2En, Ex + 2En] | | [Ex - 3En, Ex + 3En] | |
| 0 | 0.1 | 0.01 | 49.957 | 1.629 | 68.479 | 1.554 | 95.248 | 0.661 | 99.646 | 0.183 |
| 2 | 0.1 | 0.01 | 50.035 | 1.573 | 68.475 | 1.484 | 95.267 | 0.676 | 99.630 | 0.185 |
| 4 | 0.5 | 0.1 | 51.095 | 1.568 | 69.127 | 1.457 | 94.661 | 0.683 | 99.392 | 0.252 |
| 8 | 10 | 1 | 50.203 | 1.550 | 68.560 | 1.509 | 95.261 | 0.669 | 99.638 | 0.189 |
| 0 | 0.1 | 1 | 16.261 | 1.174 | 21.764 | 1.309 | 34.810 | 1.538 | 44.679 | 1.584 |
| 2 | 2 | 5 | 39.922 | 1.543 | 50.523 | 1.578 | 70.709 | 1.428 | 81.879 | 1.197 |
| 4 | 0.5 | 2 | 30.340 | 1.457 | 39.234 | 1.577 | 58.077 | 1.549 | 70.096 | 1.452 |
| 8 | 0.1 | 0.5 | 26.271 | 1.373 | 34.324 | 1.489 | 51.940 | 1.541 | 63.768 | 1.502 |

Figure 5(b) is listed in Figure 5(c), which is with an enlarged scale. As can be seen from Figure 5(c), the edge or border is *soft* or uncertain as well as frayed (referred to in [8]). In the proposed method, the default value of density factor is fixed as $\rho = 0.01$.

Furthermore, we analyse the reason to fix the parameter in detail. Given the desired height h and width w , the number of reference points can be determined according to the density factor ρ , and then the average area of each Voronoi polygon can be calculated. We set the related parameters as $h \in \{32, 64, 128, 256, 512, 1024, 1440, 1920\}$, $w \in \{32, 64, 128, 256, 512, 768, 900, 1080\}$, and $\rho \in \{0.001, 0.005, 0.01, 0.1, 0.5\}$. As can be seen from Figure 5(e), for each ρ , the average area of Voronoi polygon is increased with a larger h or w , and it shows a tendency toward stabilization. Also, we investigate the combined influence of image size (hw as one index) in log-log scale coordinates system; the similar result can be obtained from Figure 5(f). With image size varying from 1 to 10^4 , the average area of Voronoi polygon is increased with a lower ρ . However, too large polygon would result in less aesthetic effect. According to the following groups, we generate a result image with default size of 128×128 , then the area of Voronoi polygon is about 100 dots with the total area 16384 dots, and the proportion of each polygon shares about 6%. Therefore, we fix the density factor $\rho = 0.01$ by keeping a good balance between aesthetic effect and time complexity. Certainly, the density factor ρ is still an open parameter. For a given image size, one can estimate and obtain the optimal ρ according to Figures 5(e) and 5(f).

5.1.2. Group 2. In this group, we investigate the intensity factor τ , which determines the number of the removed cloud drops. Two aesthetic patterns with unsupervised decomposition are generated using $\tau = 0.0, 0.9406$, respectively. The results and the corresponding details are shown in Figure 6. With $\tau = 0.0$, none of cloud drops are removed from the set of sample points; the result image in Figure 6(a) is very delicate and with very few black backgrounds, and the Voronoi borders in Figure 6(b) show a similar effect of frosted glass. With $\tau = 0.9406$, majority of cloud drops are removed;

the result in Figure 6(c) is relatively rough, and there are several cavities near the boundaries, as shown in Figure 6(d).

In the normal cloud model, each cloud drop contributes the concept differently. In Figure 6(e), we present a visual result from a statistical view. Given $Ex = 0, En = 0.1, He = 0.01$, we generate 1000 cloud drops, and the procedure runs 1000 times. We count the number of cloud drops falling into a given interval, such as $[Ex - 0.67En, Ex + 0.67En]$, $[Ex - En, Ex + En]$, $[Ex - 2En, Ex + 2En]$, and $[Ex - 3En, Ex + 3En]$. The overall result is listed in Figure 6(e). Although there is a little difference on the results of 1000 runs between each other, the statistical feature of each run is absolutely similar. One can observe that about 50% cloud drops lie in the interval $[Ex - 0.67En, Ex + 0.67En]$, and most of cloud drops (with the proportion of about 70%) are in the interval $[Ex - En, Ex + En]$.

In addition, we repeat this procedure with three other cloud models, $Ex = 2, En = 0.1, He = 0.01$, $Ex = 4, En = 0.5, He = 0.1$, and $Ex = 8, En = 10, He = 1$. The statistical results are listed in Table 1. With a proper He/En , the distributing proportion of cloud drops in each interval is generally the same.

In fact, the Gaussian function satisfies *three-sigma rule*, and, as a consequence, normal cloud model has *3En rule*. The majority of cloud drops lie within the interval $[Ex - 3En, Ex + 3En]$, and, specifically, cloud drops within the interval $[Ex - 0.67En, Ex + 0.67En]$, called the backbone elements, only account for 22.33% of the universe set but contribute 50% of the cloud model. Then, cloud drops within the interval $[Ex - En, Ex + En]$ make up 33.33% of the universe and contribute 68.26% of the cloud concept. For these cloud drops, the threshold of the certainty degree is 0.6065, that is, the intensity factor in the proposed algorithm. Additionally, about 11.11% of cloud drops are within $0.33En$ away from Ex and contribute 25.86% of the cloud concept; then the intensity factor is 0.9406. Similarly, other cases can be also calculated. To obtain the universal aesthetics, the default value of intensity factor is fixed as $\tau = 0.6065$ in the proposed method.

While with an extreme He/En , the distributing proportion of cloud drops is quite different, as shown in the latter four rows of Table 1. In this condition, the corresponding

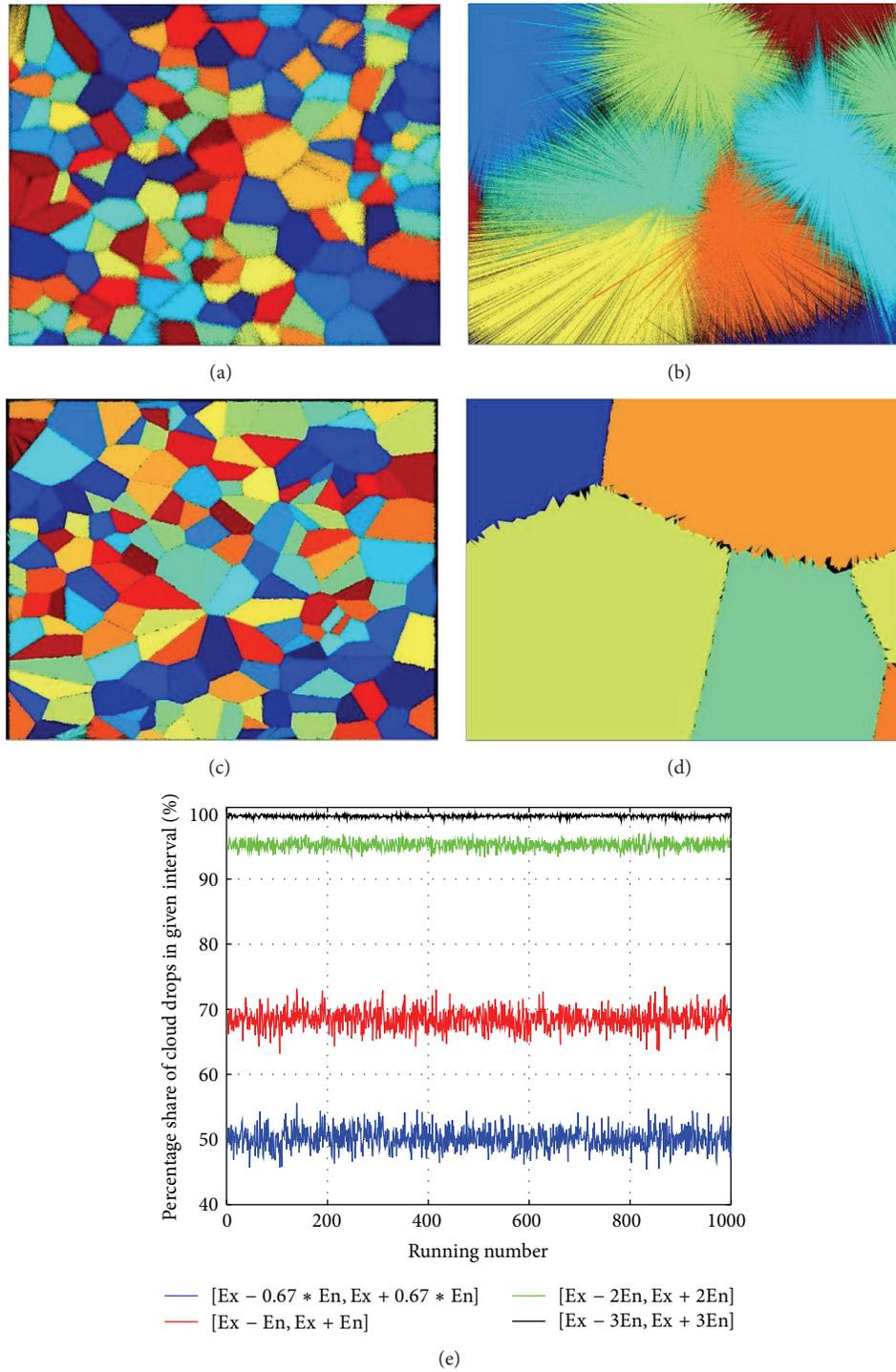


FIGURE 6: Rendering with various degrees: (a) $\tau = 0.0$, (b) the detail with an enlarged scale for (a), (c) $\tau = 0.9406$, (d) the detail with an enlarged scale for (c), and (e) statistical result on cloud drops and their contribution.

cloud model is atomized. In Group 5, we will further discuss this condition.

5.1.3. *Group 3.* In this group, we try to provide a study on the color schemas for the proposed method. All the above results used the color schema of the Jet colormap and the black

background, which is default for the unsupervised decomposition. Besides that, the summer colormap, the gray colormap, and the white background are also tested. Three aesthetic patterns are generated, as shown in Figure 7, and the other three results (EPS results) are attached in the Supplementary Material available online at <http://dx.doi.org/10.1155/2016/9837123>.

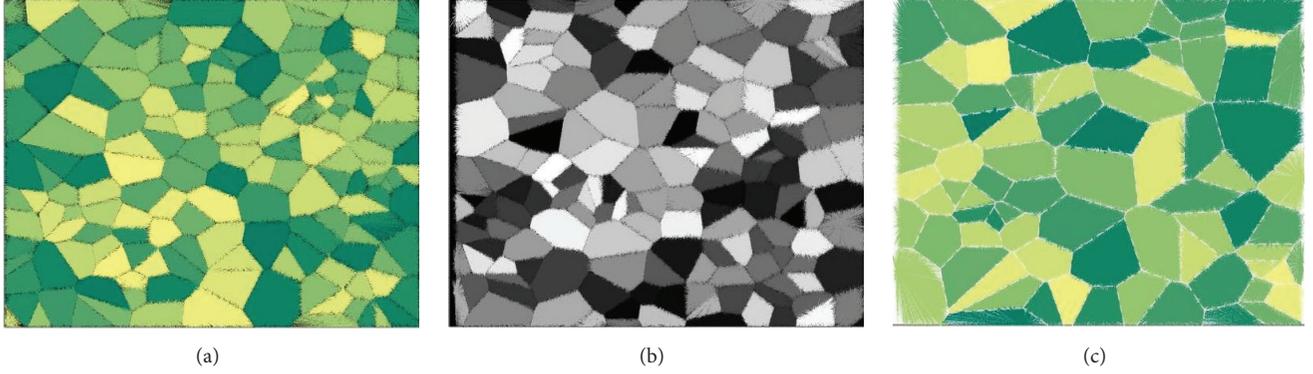


FIGURE 7: Rendering with different color schemes: (a) the summer colormap, (b) the gray colormap, and (c) the summer colormap with white background.

In most cases of image-guided decomposition, the distribution of Voronoi polygons is dense, and a too bright color is not conducive to the rendering of image objects; then the color schema is with the summer colormap and the black background, which is the default for the image-guided decomposition. In general, the experimental results suggest that the color schema is more important to image-guided decomposition than unsupervised decomposition. Certainly, any other color schemas are also optional, and it is an open problem.

5.1.4. Group 4. In this group, we investigate the number of the cloud drops n , which is also very important to the time performance of the proposed method, as mentioned above. Two aesthetic patterns with unsupervised decomposition are generated using $n = 10, 5000$, respectively. The results and the corresponding details are shown in Figure 8. With $n = 10$, sample points cannot well approximate the lines; the filling result in Figure 8(a) is too rough, and there are larger void spaces near the border, as the black patches in Figure 8(b). With $n = 5000$, the number of sample points in Figure 8(c) is excessive, the overlapping between each couple of polygons is very common, and then the uncertainty of the lines and polygons is weakened, as shown in Figure 8(d). The uncertain line and polygon would become the thick and “hard” one in the case of very large n .

Comparing Figure 8(b) with Figure 6(d), they are similar in aesthetics effect, although the former is used with a less number of the cloud drops and the latter is with a higher intensity factor. In fact, most of the cloud drops and the sample points are still removed even given a higher intensity factor and a larger number of the cloud drops, and it is equivalent to the case of a less number of the cloud drops.

Taking the above endpoints of a line segment as an example, that is, $\mathbf{P}_1 = (0, 1)$ and $\mathbf{P}_2 = (4, 9)$, the reference point is $\mathbf{P}_r = (5/4, -5/2)$, and we generate cloud drops with varied cloud model (the entropy $[1/px_r, 1/py_r]^T$ and the hyper-entropy $[0.1/px_r, 0.1/py_r]^T$). The procedure is repeated 1000 times, and the number of cloud drops increases 5 iteration by iteration. In other words, the number of cloud drops is varied from 1 to 5000. For each iteration, we record

the mean squared error (MSE) between the cloud drops and the real values in the line $\mathbf{P}_1\mathbf{P}_2$. The results are shown in Figure 8(e), and for the reference, the average MSE is also drawn. With the number of cloud drops less than 500, the MSE values are fluctuated violently, which leaves the uncertain process in an unstable state. Meanwhile, a larger number of cloud drops would result in to be more time-consuming according to the time complexity. Thus, a modest n about 500 to 1000 is beneficial. In the proposed method, the default value of the number of cloud drops is fixed as $n = 500$.

5.1.5. Group 5. In this group, we investigate the numerical characteristics of the cloud model, and an improvement is added into the proposed algorithm. Two aesthetic patterns with unsupervised decomposition are generated; the results are shown in Figure 9. Given both the entropy and the hyper-entropy $[0, 0]^T$, the cloud model has degenerated into a normal distribution; the uncertainty is completely evaporated and equivalent to the certain and accurate processing, as the traditional method does. From Figure 9(a), the aesthetic effect is identical with the classical Voronoi art.

From another perspective, cloud drops present a generalized Gaussian distribution when $He < En/3$; the corresponding cloud model is generally suitable for the representation of qualitative concept. Conversely, cloud drops appear in atomized state in case of $He > En/3$, the samples seriously deviate from Gaussian distribution, and then a cloud concept is difficult to reach and achieve. As shown in the latter four rows of Table 1, the distributing proportion of cloud drops is quite different from a normal cloud model. Thus, we should avoid this condition using parameter setting.

We present an example with both the entropy and the hyper-entropy $[5, 5]^T$, the cloud model has satisfied the atomized condition, and the result image has none of aesthetics. Sample points cannot effectively approximate the Voronoi polygons, and it fails to represent lines and fill polygons with uncertainty, as shown in Figure 9(b). From this perspective, an additional step should be introduced into Step 4 in Section 4, besides the intensity factor of the certainty degree, that is, filtering cloud drops and sample points outside the polygons. This step ensures that the proposed algorithm

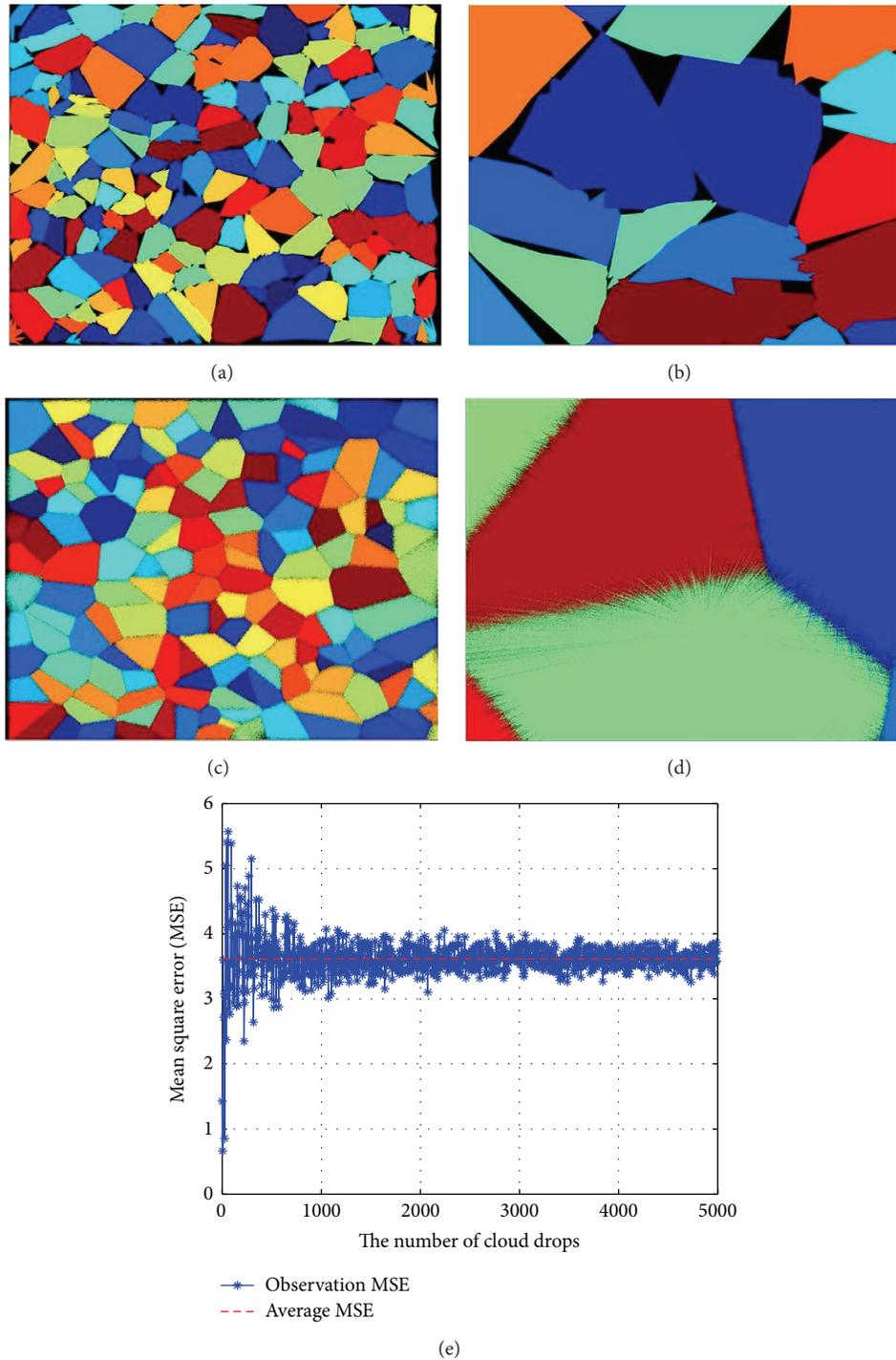


FIGURE 8: Rendering with various number of cloud drops: (a) $n = 10$, (b) the detail with an enlarged scale for (a), (c) $n = 5000$, (d) the detail with an enlarged scale for (c), and (e) mean squared error of cloud drops.

is robust even if involving invalid values of the entropy or the hyper-entropy. As a result, another aesthetic style is generated.

Furthermore, we generate another aesthetic pattern with the entropy $[0.1, 0.1]^T$ and the hyper-entropy $[0.01, 0.01]^T$; the result is shown in Figure 9(c). With the constant and

nonzero values of entropy and hyper-entropy, the rendering result is low-aesthetic, although the condition of the generalized Gaussian distribution is satisfied. Obviously, the result image is asymmetric, and its left is more delicate. This is because k_i and b_i are generated by constant numerical characteristics of cloud model, and these two parameters

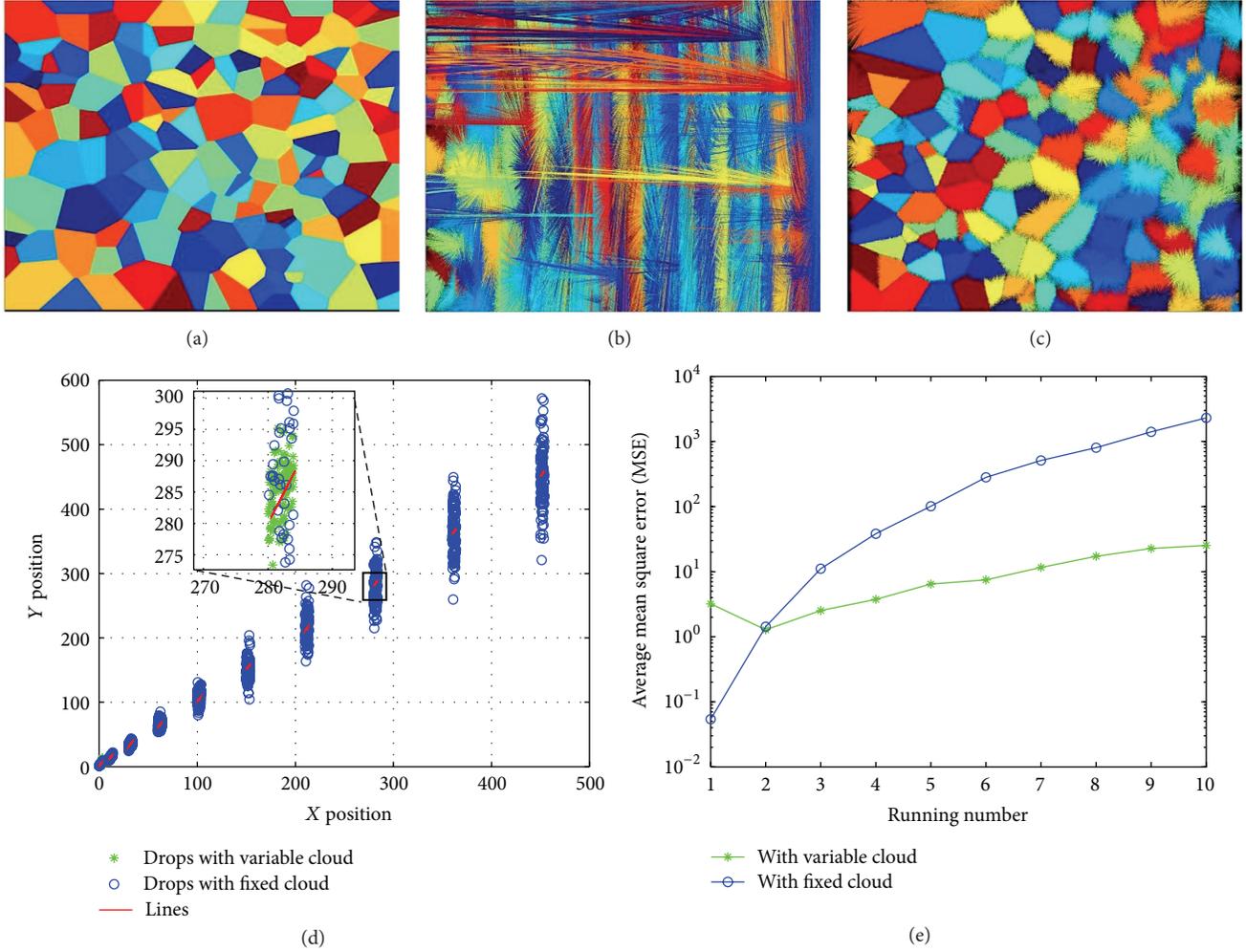


FIGURE 9: Rendering with various parameters of cloud model: (a) the degradation of cloud model, (b) the atomization of cloud model, (c) the constant and nonzero values for En, He, (d) uncertain line with fixed and variable cloud models, and (e) average MSE of fixed and variable models.

approximate the real values at different levels, and then the left sample points with smaller lx_i would be the lower deviation from the real line, according to $ly_i = k_i lx_i + b_i$. Otherwise, the right sample points would be the higher deviation. Thus, the right half plane appears rougher than the left. To void this problem, at least one of entropy and hyper-entropy is fixed as a reduction function related to the positions of reference points.

Considering $\mathbf{P}_1 = (0, 1)$ and $\mathbf{P}_2 = (4, 9)$ with the reference point $\mathbf{P}_r = (5/4, -5/2)$ as the starting, the expected values are calculated by (2), and then we generate 1000 cloud drops; half of them are from fixed cloud models with the entropy $[0.1, 0.1]^T$ and the hyper-entropy $[0.01, 0.01]^T$, and the other half are from variable cloud models with the entropy $[1/px_r, 1/py_r]^T$ and the hyper-entropy $[0.1/px_r, 0.1/py_r]^T$. This procedure is repeated 10 times with an incremental position, and the increment of one step is 10 on both x - and y -axis. Then, from left to right and from bottom to top, there are ten lines by ten runs. The positions of cloud drops

are drawn in Figure 9(d) as well as the lines. Only the lines and the drops from fixed cloud models are visible, while the drops from variable cloud models are generally not visible. For further observation, we also show an enlarged scale for the 8th line, and, in this state, we could vaguely see the drops from variable cloud models. That is to say, the drops from variable cloud models cluster around the represented line, and they are covered by those from fixed cloud models, which seriously spread along the line. With the increment of X and Y position, the distribution on drops from fixed cloud models is more dispersed. Additionally, we list the average MSE values between cloud drops and real values of lines. As can be seen from the Y -log system in Figure 9(e), with a smaller coordinate values, the average MSE value of fixed cloud model seems to be less than that of variable cloud model, while, with the increase of coordinate values, variable cloud models achieve absolutely smaller average MSE values. When the X and Y position is about 500, the fixed cloud models produce 1000 times average MSE value as much as

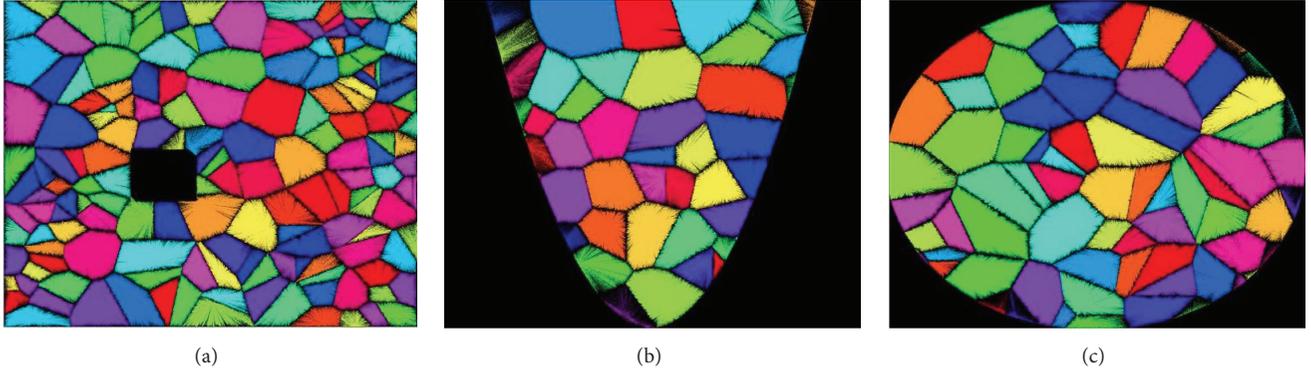


FIGURE 10: Rendering with various shapes: (a) the rectangle with an inside hole, (b) the area with a parabolic outer boundary, and (c) a circular region.

that of variable cloud model. Therefore, the default setting in the proposed method is with the entropy $[1/px_r, 1/py_r]^T$ and the hyper-entropy $[0.1/px_r, 0.1/py_r]^T$.

5.1.6. Group 6. In this group, we render three Voronoi-based images with various shapes. For all of the patterns, the size is as $h = w = 128$. The first one is a rectangle and its central part is blown-up, determined by $(40, 70)$, $(40, 50)$, $(60, 50)$, $(60, 70)$. The decomposition result is listed in Figure 3(c), and the final pattern is shown in Figure 10(a), with a high and aesthetical quality. The second is an irregular area surrounded by the parabola $y = 0.05(x - 64)^2 + 1$ and the line $y = 128$. The output in Figure 10(b) remains generally satisfactory, with only fewer faulty borders because of the shape constraint. The last is a disk determined by $(x - 64)^2 + (y - 64)^2 \leq 64^2$, and the result is also acceptable, as shown in Figure 10(c). In summary, these aesthetic patterns indicate that the proposed algorithm can tolerate various shape constraints and generate a relatively comfortable visual effect. In theory, any area with any shape constraint can obtain similar stylized image through the proposed method, which provides possible applications in various fields, such as tessellation-based image stylization, image mosaic, and Voronoi art.

5.1.7. Group 7. Using the proposed algorithm, four images are processed, and the result images are listed in Figure 11, attached by the original images. As shown in Figures 11(a) and 11(d), the positions of the background pixels are as the coordinates of the reference points, while those of the object pixels are in Figures 11(c) and 11(b). To be clear, there is the result of the IE logo image in Figure 11(a), whose half-tone image is as mentioned above in Figure 3(b), and another result is attached in the Supplementary Material. Summarily, these results are favourable, and all of them demonstrate the high aesthetics effect, which is similar to the stroke based painterly rendering, but different from the existing methods.

5.2. Visual Comparisons. This subsection provides a qualitative comparison of our output against the relative methods. All of methods, including the traditional method, the FCD

TABLE 2: Average running times for various image sizes.

| Size | | Running times (s) | | |
|------|------|-------------------|------------------|----------|
| h | w | Ours | Classical method | FCD |
| 32 | 32 | 0.386 | 0.245 | 7.533 |
| 64 | 64 | 0.765 | 0.295 | 34.209 |
| 128 | 128 | 2.412 | 0.418 | 265.404 |
| 256 | 256 | 8.351 | 1.006 | 2478.647 |
| 512 | 512 | 34.199 | 4.416 | >14400 |
| 1024 | 768 | 109.572 | 22.586 | >14400 |
| 1440 | 900 | 199.271 | 51.778 | >14400 |
| 1920 | 1080 | 352.121 | 115.754 | >14400 |

method [8], and ours, are implemented in Matlab and performed on a 2.4 GHz Core i7 PC with 8 GB RAM. As shown in Figures 12(b) and 12(e), the Voronoi-based rendering by the classical method is very crisp, but without uncertainty, while the results by our method and the FCD method are soft and rough, as shown in Figures 12(a) and 12(c). Even so, there is the obvious difference between them. Result in Figure 12(f) by the FCD method appears in rotational symmetry, where each reference point is the center of symmetry. Although it is regular and organized, the handling of the uncertainty is too rigid, and there are signs that the result is man-made and artificial. Compared with the FCD method, our method in Figure 12(d) handles the uncertainty using cloud model and makes the result closer and more harmonious with the nature. In fact, the user study in the following is also indicating that our results are more like an effect with the frayed or cracked soil. However, we do not mainly focus on evaluating aesthetics of the generated patterns, and our method still captures the uncertainty better than the existing methods.

5.3. Time Performance Analysis. In this subsection, we investigate the running time of the proposed method. We run 10 times to get mean values for each group of parameters. The running times are listed in Table 2. With the increase of the size of the input image, the generation time of all methods have increased. From the time cost's point of view,

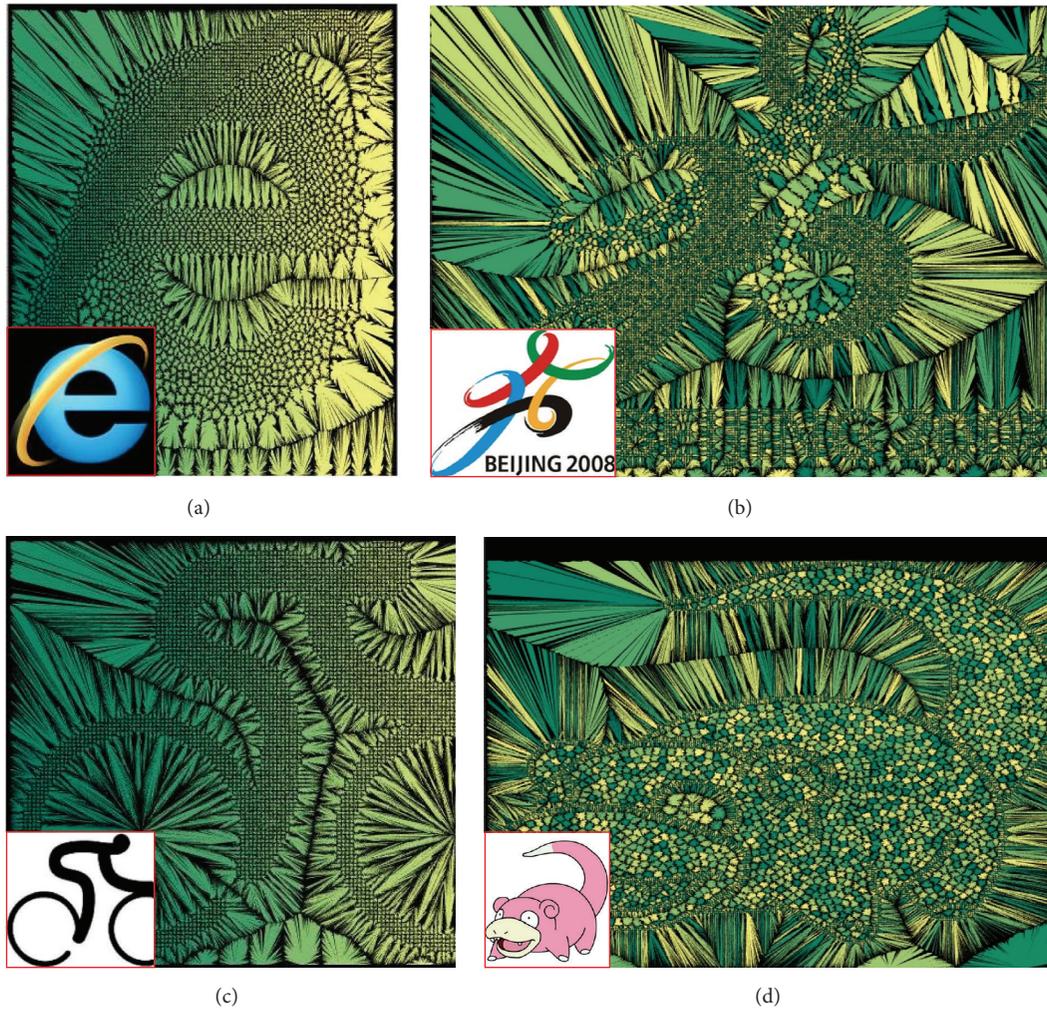


FIGURE 11: Voronoi stylization for images: (a) the IE logo, (b) the Beijing Olympic logo, (c) the Olympic cycling logo, and (d) a cartoon picture.

the classical method is the fastest, but noneffective, and none of the uncertainty is handled, which can be seen from the above sections. For a novel visual effect, our method and the FCD method spend much time to replace the rendering operation in the traditional method, and the running time increases inevitably. The FCD method is the most time-consuming, even its time cost is about 50 times as much as the cost of the proposed method. With a larger size of the input image, our method is more time-saving than the FCD method. Specifically, the FCD method cannot obtain the output spanning 4 hours in the case of 512×512 and we label the time > 14400 seconds as shown in Table 2.

The results of Table 2 are also supported by the theoretical analysis on the time complexity. The FCD method takes the pixel as the basic processing unit, the outer loop iterates hw times, and the innermost loop to find the reference point for each pixel scans each Voronoi polygon, which costs the time $O(N)$ at least. Thus, the time cost of the FCD method is about $O(Nhw)$ ($O(\rho h^2 w^2)$ for the unsupervised decomposition), which is the quadratic complexity of the size of the input image, and more time-consuming than ours $O(hw)$.

To further investigate the time performance, we provide an analysis on the change of time costs varied with image size (i.e., hw). The original data is from Table 2. Using the `polyfit()` function in Matlab, we try to construct a straight line that has the best fit to these data points. The results are shown in Figure 13, and the fitting result of the FCD method has been moved since it is far from a straight line, suffers a startling deviation, and seriously influences the illustration of the results by ours and the classical method. Clearly, our method shows a good performance to line fitting. Theoretically, in some extension, it has a significant linear correlation with the image size.

It would be specially mentioned that hw here corresponds to the size of the input image for image-guided decomposition, but not the image resolution or size of the generated result. Our method uses the technique of vector drawing and can be saved as various vector format, which can be chosen by users; for example, the scanning precision of EPS (Encapsulated PostScript) reaches 600 DPI (Dots Per Inch). In a way, the proposed method is efficient from the view of running time and can be approximatively satisfied by

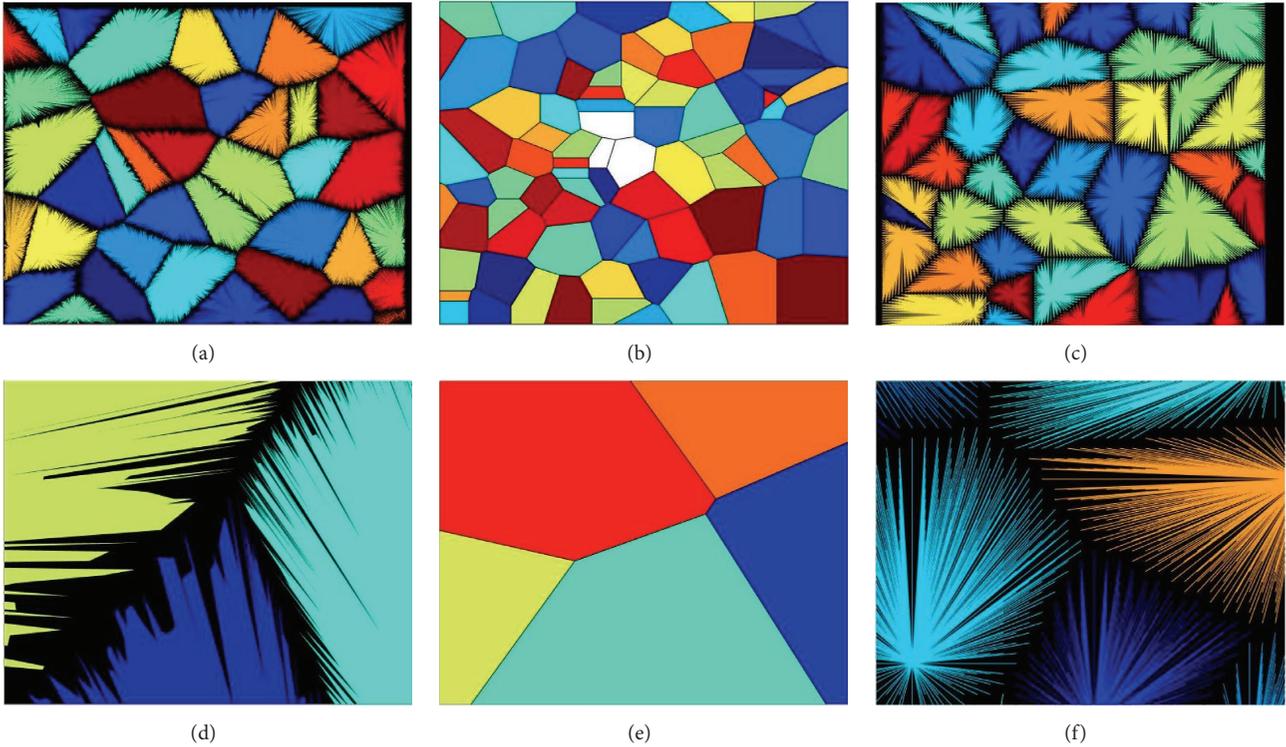


FIGURE 12: Results by various methods: (a) ours, (b) the classical method, (c) the FCD method, (d) the detail with an enlarged scale for (a), (e) the detail with an enlarged scale for (b), and (f) the detail with an enlarged scale for (c).

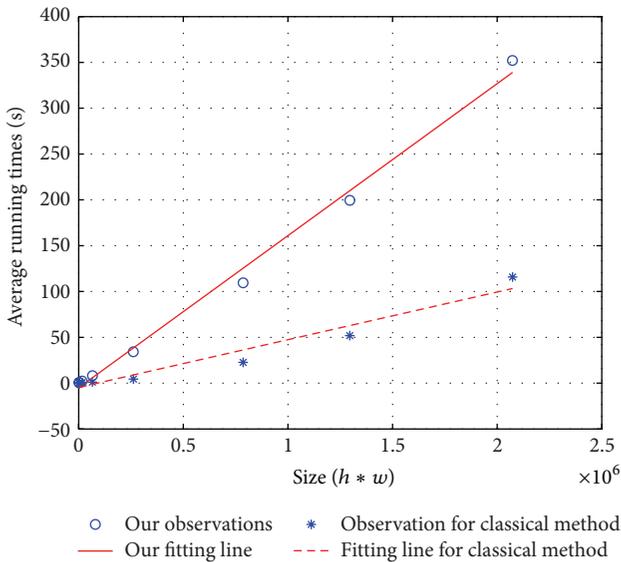


FIGURE 13: Line fitting on time performance of our method and the classical method.

the need of the real-time applications, because our method runs with less time cost and the generated visual results are acceptable.

5.4. *Quantitative Comparisons.* In this subsection, we use five indexes to provide a quantitative comparison, including

TABLE 3: Quantitative evaluation results.

| Index | Ours | Classical method | FCD |
|------------------|---------------|------------------|--------|
| Index_B | 0.8638 | 0.6696 | 0.7265 |
| Index_F | 0.7006 | 0.8673 | 0.9276 |
| Index_S | 0.8775 | 0.7166 | 0.7634 |
| Index_G | 0.6881 | 0.3863 | 0.6383 |
| Index_K | 0.7581 | 0.6693 | 0.5565 |

Benford's law index_B [27], fractal dimension index_F [28], Shannon entropy index_S [29], global contrast factor index_G [30], and Kolmogorov complexity index_K [31]. The result images in Section 5.2 are involved, and the total number is 200, in which our method, the classical method, and the FCD method account for 40%, 40%, and 20%, respectively. Because of randomness, we record the score of each image and then average them according to the used method. For comparison purposes, each row of the score of each measure is normalized.

The quantitative evaluation results are listed in Table 3. Benford's law index_B is the measure of the distribution of intensity of pixels. The index_B score gradually reduces in the order of the classical method, the FCD method, and our method, and the corresponding average deviation becomes larger and larger. Images with a higher index_F were considered complex, and images with a lower one were more uncertain. Our results show the uncertainty with the lowest index_F. The Shannon entropy index_S rewards images with

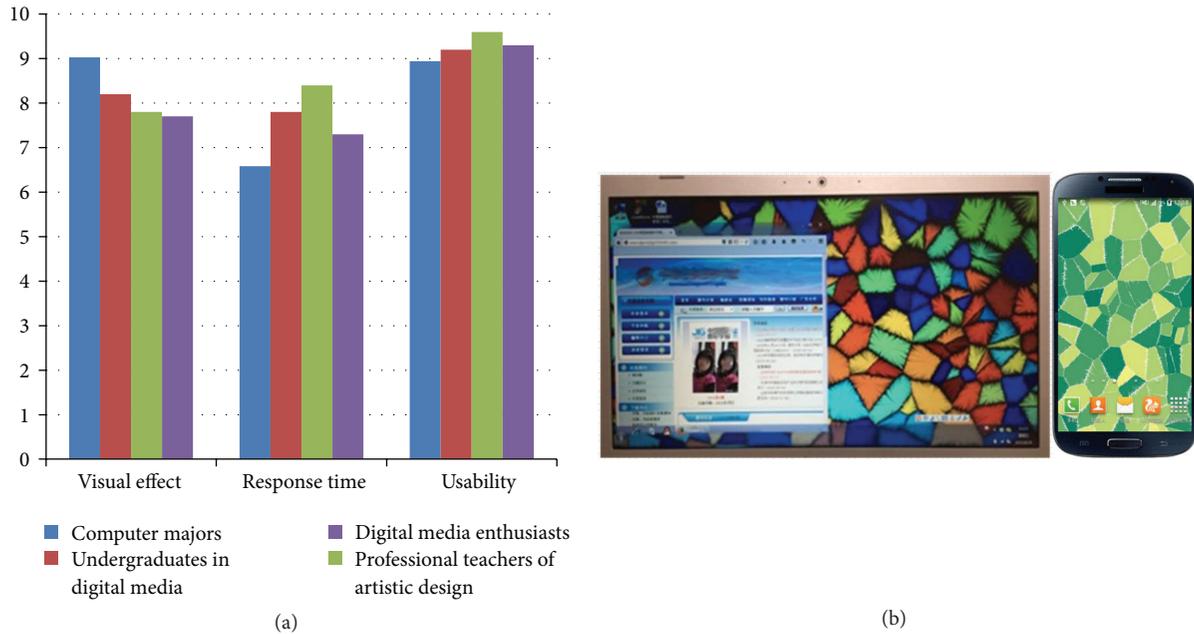


FIGURE 14: User survey results: (a) the scores from user survey and (b) the screenshot using the proposed aesthetic image.

a uniform distribution of brightness values, and our results obtain the highest of all, while the classical results are the lowest. From this perspective, our method generates the patterns with more uncertainty. The global contrast factor calculates and values contrast on various resolutions of an image, and a lower contrast index $_C$ reflects a lower aesthetic effect. The index $_C$ score of the proposed method ranks first, followed by the FCD method and the classical method, and it is indicated that our method generates the results with the highest aesthetics. The Kolmogorov complexity index $_K$ and the Shannon entropy index $_S$ complement each other; their results reflect the similar meaning. Overall, our method is certainly more efficient from the perspective of uncertain or frayed visual effect, since it achieves a higher index $_B$ (the distribution of intensity of pixels is more deviated from Benford's law), a lower index $_F$ (more uncertain), a higher index $_S$ and index $_K$ (nonuniform distribution of brightness values), and a higher index $_G$ (higher aesthetic effect).

However, quantitative measure of aesthetics remains a challenging task, although for specific image applications some recent progress has been made in this direction. Still, the main purpose by the quantitative comparison in Table 3 is to assess the difference between pair of the aesthetic patterns generated by different methods, and it should not be strictly limited to scores of the various styles. In other words, the quantitative measure of aesthetics is a necessary and beneficial complement to the visual comparison and the user study. For a full evaluation of the new artistic preprocessing technique, we design a user study involving both amateur users with some art knowledge and professional art students, as the following subsection.

5.5. The User Study. Humans are an integral part of defining what is aesthetic; thus we employ a user study to assess the

performance in this subsection. Using the provided GUI, the users control the type of Voronoi decomposition and fix the parameters. Each user is required to obtain 50 images independently. Then, the study asked the users to score each image according to a sliding scale of measure value, using a Likert scale in the range [0 10], and 10 means a perfect result. A total of 100 users participated in this investigation; they volunteered from our university, and there are 87 computer majors, 5 undergraduates in digital media, 5 professional teachers of artistic design, and 3 digital media enthusiasts. The evaluation covers threefold: (1) visual effect: we asked the user to score the aesthetic feeling of each result, and 10 means the most aesthetic. (2) Response time: each participant is asked to score the satisfaction of running time, and 10 means the most satisfied. (3) Usability: the users choose a score to measure the possibility of the generated patterns used in real applications, and 10 means the most possible.

Figure 14(a) shows a summary of the survey results, and Figure 14(b) shows two most pleased results, used in computer background and mobile wallpaper. The computer majors score the highest visual effect, and the average score of the visual effect is more than 7.5. For the subjectivity of the aesthetics problem, our work is only based on the opinions of our colleges and students who have seen our patterns. The users are generally satisfied with the performance of the proposed method, and majority of them said that our results were looking nicely and more like an effect with the frayed or cracked soil. The response time cannot perfectly meet the most demanding of the computer majors, that mark the lowest for this measure. The average score of the response time is more than 6.5, and the professional teachers of artistic design show that the response time of the proposed iVPC method is acceptable, since they often encountered the longer wait at work. The average score of the usability is above 8,

and the computer majors are pleased to take these results as wallpapers (see Figure 14(b)), and the others also think the generated patterns are helpful to their daily designs. In summary, the user study shows a good performance of the proposed method.

6. Summary and Conclusion

In this paper, an uncertainty algorithm based on cloud model for image-guided Voronoi aesthetic patterns has been proposed. As a computational intelligence tool, cloud model handles the uncertainty more completely and more freely, and it cannot be considered as randomness compensated by fuzziness, fuzziness compensated by randomness, second-order fuzziness, or second-order randomness. To obtain the default parameters, we conduct seven groups of experiments to test the proposed method. Using both visual and quantitative comparisons, we prove the efficacy of the proposed method using two groups of experiments. Compared with the related methods, experimental results show that the Voronoi-based aesthetic patterns with *soft* borders can be generated by using the new technique. Also, the proposed iVPC algorithm has better performance on running time, and its time complexity is approximately linear related to the size of the input image. The real engineering applications using the proposed algorithm, such as home decoration and clothing design, are under the investigation and implementation and would be reported in the future. Nevertheless, we still see a good potential that the soft rendering technique could be applied to diagrams from computer-based visualization and graphics.

There are a couple of issues that should be considered in the proposal: (1) the proposed method used a considerable number of line drawing, and the vector output consumes more file storage space. For the input image with size of 256×256 , the output file needs about 30MB. Thus, how to generate other formats, for example, SVG (Scalable Vector Graphics), and reduce the file space is one of the useful extensions. (2) The proposed method is implemented in Matlab, and other languages for engineering applications should be employed, for example, C++ and Java. Then, how to integrate with the existing software is another feasible direction. The extension of the technique is currently under investigation and will be reported later.

Competing Interests

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

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (under Grant no. 61402399), by the National Key Basic Research and Development Program (no. 2012CB719903), by the Foundation for Distinguished Young Teachers in Higher Education of Guangdong, China (no. YQ2014117), and by the Foundation of Humanities and Social Sciences Research in Ministry of Education, China (no. 14YJCZH161).

References

- [1] M. David, "Image-guided fracture," in *Proceedings of the Graphics Interface Conference*, pp. 219–226, Canadian Human-Computer Communications Society, Victoria, Canada, May 2005.
- [2] F. Hoenic, "Defining computational aesthetics," in *Proceedings of the 1st Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging*, pp. 13–18, 2005.
- [3] P. Barile, V. Ciesielski, K. Trist, and M. Berry, "Animated drawings rendered by genetic programming," in *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO '09)*, pp. 939–948, New York, NY, USA, 2009.
- [4] D. L. Atkins, R. Klapaukh, W. N. Browne, and M. Zhang, "Evolution of aesthetically pleasing images without human-in-the-loop," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '10)*, pp. 1–8, 2010.
- [5] E. Den Heijer and A. E. Eiben, "Investigating aesthetic measures for unsupervised evolutionary art," *Swarm and Evolutionary Computation*, vol. 16, pp. 52–68, 2014.
- [6] C. M. Fernandes, A. M. Mora, J. J. Merelo, and A. C. Rosa, "Photorealistic rendering with an ant algorithm," in *Computational Intelligence: International Joint Conference, IJCCI 2012 Barcelona, Spain, October 5–7, 2012 Revised Selected Papers*, vol. 577 of *Studies in Computational Intelligence*, pp. 63–77, Springer, New York, NY, USA, 2015.
- [7] S. Kim, R. Maciejewski, A. Malik, Y. Jang, D. S. Ebert, and T. Isenberg, "Bristle maps: a multivariate abstraction technique for geovisualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 9, pp. 1438–1454, 2013.
- [8] B. Michael, V. Corinna, and W. Daniel, "Frayed cell diagrams," in *Proceedings of the ACM Workshop on Computational Aesthetics (CAe '14)*, pp. 93–96, British Columbia, Canada, August 2014.
- [9] F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, 1991.
- [10] C. S. Kaplan, "Voronoi diagrams and ornamental design," in *Proceedings of the 1st Annual Symposium of the International Society for the Arts, Mathematics, and Architecture*, pp. 277–283, 1999.
- [11] L. Golan, "Segmentation and symptom created for british arts quarterly zoo," <http://www.flong.com/projects/zoo>.
- [12] M. David, "A stained glass image filter," in *Proceedings of the 14th Eurographics Workshop on Rendering*, pp. 20–25, Eurographics Association, Leuven, Belgium, 2003.
- [13] W. Brian, "Determining an aesthetic inscribed curve," in *Proceedings of the 8th Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging, Eurographics Association*, pp. 63–70, Annecy, France, 2012.
- [14] M. Jooyandeh, A. Mohades, and M. Mirzakhah, "Uncertain voronoi diagram," *Information Processing Letters*, vol. 109, no. 13, pp. 709–712, 2009.
- [15] D. Sun and Z. Hao, "Group nearest neighbor queries based on voronoi diagrams," *Computer Research and Development*, vol. 47, no. 7, pp. 1244–1251, 2010.
- [16] X. Wang, H. Zhang, Q. Fang, Y. Ge, and Z. Wang, "Research on probabilistic Voronoi model and algorithm for coverage in WSN," *Chinese Journal of Sensors and Actuators*, vol. 25, no. 5, pp. 702–706, 2012.
- [17] T. Isenberg, "Visual abstraction and stylisation of maps," *The Cartographic Journal*, vol. 50, no. 1, pp. 8–18, 2013.

- [18] D. Li and Y. Du, *Artificial Intelligence with Uncertainty*, Chapman & Hall, Boca Raton, Fla, USA, 2007.
- [19] D. Li, C. Liu, and W. Gan, "A new cognitive model: cloud model," *International Journal of Intelligent Systems*, vol. 24, no. 3, pp. 357–375, 2009.
- [20] K. Qin, K. Xu, F. Liu, and D. Li, "Image segmentation based on histogram analysis utilizing the cloud model," *Computers and Mathematics with Applications*, vol. 62, no. 7, pp. 2824–2833, 2011.
- [21] G. Wang, C. Xu, and D. Li, "Generic normal cloud model," *Information Sciences*, vol. 280, pp. 1–15, 2014.
- [22] T. Wu, J. Xiao, K. Qin, and Y. Chen, "Cloud model-based method for range-constrained thresholding," *Computers & Electrical Engineering*, vol. 42, pp. 33–48, 2015.
- [23] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.
- [24] D. Oliver and I. Tobias, "Halftoning and stippling," in *Image and Video-Based Artistic Stylisation*, pp. 45–61, Springer, Berlin, Germany, 2013.
- [25] R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial grey scale," in *Proceedings of the Society of Information Display*, pp. 75–77, 1976.
- [26] S. Jakob, Constraining the vertices of a voronoi decomposition, <http://cn.mathworks.com/matlabcentral/fileexchange/34428voronoilimit>.
- [27] J.-M. Jolion, "Images and benford's law," *Journal of Mathematical Imaging and Vision*, vol. 14, no. 1, pp. 73–81, 2001.
- [28] B. Spehar, C. W. G. Clifford, B. R. Newell, and R. P. Taylor, "Universal aesthetic of fractals," *Computers and Graphics*, vol. 27, no. 5, pp. 813–820, 2003.
- [29] J. Rigau, M. Feixas, and M. Sbert, "Informational aesthetics measures," *IEEE Computer Graphics and Applications*, vol. 28, no. 2, pp. 24–34, 2008.
- [30] K. Matkovic, L. Neumann, A. Neumann et al., "Global contrast factor—a new approach to image contrast," in *Proceedings of the 1st Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging*, pp. 159–168, 2005.
- [31] J. Rigau, M. Feixas, and M. Sbert, "Conceptualizing birkhoff's aesthetic measure using shannon entropy and kolmogorov complexity," in *Proceedings of the 3rd Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging*, pp. 105–112, 2007.

Research Article

Learn to Rank Images: A Unified Probabilistic Hypergraph Model for Visual Search

Kaiman Zeng, Nansong Wu, Arman Sargolzaei, and Kang Yen

Department of Electrical and Computer Engineering, Florida International University, 10555 West Flagler Street, Miami, FL 33174, USA

Correspondence should be addressed to Kaiman Zeng; kzeng001@fiu.edu

Received 5 March 2016; Revised 11 June 2016; Accepted 16 June 2016

Academic Editor: Daniel Zaldivar

Copyright © 2016 Kaiman Zeng 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 visual search systems, it is important to address the issue of how to leverage the rich contextual information in a visual computational model to build more robust visual search systems and to better satisfy the user's need and intention. In this paper, we introduced a ranking model by understanding the complex relations within product visual and textual information in visual search systems. To understand their complex relations, we focused on using graph-based paradigms to model the relations among product images, product category labels, and product names and descriptions. We developed a unified probabilistic hypergraph ranking algorithm, which, modeling the correlations among product visual features and textual features, extensively enriches the description of the image. We conducted experiments on the proposed ranking algorithm on a dataset collected from a real e-commerce website. The results of our comparison demonstrate that our proposed algorithm extensively improves the retrieval performance over the visual distance based ranking.

1. Introduction

Ranking plays an essential role in a product search system. Given a query, candidate products should be ranked according to their distance to the query. The effectiveness of the product search system is evaluated by its ranked search results, for example, in the form of precision or recall. In addition, the efficiency of a system is evaluated by its running time of a query. The best scenario is that the system returns a series of relevant products at the top of retrieved results. However, in certain cases, even if a system finds the particular relevant product, it is still considered as ineffective for the reason that the retrieved product is not present in the top list but is buried in a number of irrelevant results. In order to compensate for this rank inversion issue, the automated learning techniques and the skills of users are utilized to improve the representation of the query product.

A natural extension of such add-value process is to request users to label the returned results as relevant or irrelevant, which is called relevance feedback (RF). However, in reality users are neither willing to initialize a query by labeling retrieval metadata and samples nor willing to

give feedback of the retrieved results, since these methods make the retrieval procedure inconvenient. Therefore, the insufficient user-labeled images undermine the prospect of supervised learning methods in the visual content-based image retrieval (CBIR) field. A promising and relatively unexplored research direction is to exploit transductive or semisupervised learning, among which graph-based methods [1–4] have demonstrated their effectiveness in image retrieval and therefore received increasing attention. In the graph-based methods, a graph is built on the image dataset and each image is considered as a vertex in the graph. An edge and its weight are defined between two images according to a certain relationship definition. For example, the edge can be defined as the images visual similarity. The weight is formulated by the visual distance between any two image vertices. Then, the ranking can be formulated as a random walk on the graph [1, 5], or an optimization problem [2]. However, these graphs, created in pairs, cannot sufficiently show the relations among images. Hypergraph is introduced to the CBIR field. Hypergraph is a generalization of a simple graph. In a hypergraph, an edge, called hyperedge, can connect any number of vertices; it is a nonempty set of vertices.

A probabilistic hypergraph is proposed in [3] for image retrieval ranking. The hyperedge is formed by a centroid image and its k -nearest neighbors based on their visual similarity. Gao et al. [4] propose a hypergraph learning algorithm for the social image search, where the weight of hyperedges, representing the impact of different tags and visual words, is automatically learned with a set of pseudopositive images. Their research [3, 4] proves the effectiveness of hypergraph learning in solving ranking problems. However, they fail to establish correlations between images visual content and textual content. They both built the hypergraph model on images visual content or textual content independently. As a result, the search must start with a user assigned keyword. In order to solve this issue and minimize user's efforts in query, we take the product rich textual information into the visual model of image retrieval and propose a novel hypergraph-based transductive algorithm for ranking of visual product search. We design a unified probabilistic hypergraph to model multiple types of features of the products and explore the implicit relations among various visual and textual features.

The contributions of this paper are summarized in three aspects as follows. First, hypergraph is proposed to represent a commercial product image dataset. The relations between visual and textual features of these images have been explored. Second, a new product retrieval framework for the product search is designed. Third, a novel strategy of starting a query is created. Since relations between visual features and textual features have been established in a specific unified probabilistic hypergraph, problems that lack user-labeled query keywords can be solved via transductive inference on the hypergraph.

The paper is organized as the following outline. In Section 2, we have a literature review of recent ranking techniques. Section 3 discusses the design of the proposed unified hypergraph ranking algorithm. In Section 4, several retrieval experiments are conducted on an apparel dataset and compared with conventional CBIR ranking methods. Finally, we conclude with the proposed ranking scheme and discuss future works in Section 5.

2. Related Work

Ranking and hypergraph learning are the two research fields related to our work. These two topics have received intensive attention in information retrieval and machine learning. The conventional image ranking is developed from textual retrieval. The ranking model is defined based on the bag of words, for example, Best Match 25 [6], the Vector Space Model [7], and the Language Modeling for Information Retrieval [8]. Another type of ranking model is based on hyperlink analysis, such as Hyperlink-Induced Topic Search [9], PageRank [10], and its variations [11–13].

In the CBIR systems, the ranking is commonly obtained from the similarity measure of adopted visual features. One type of similarity measures is calculated from Minkowski distance, City-block distance, infinity distance, and cosine distance. They are usually called Minkowski and standard

measures. Statistical measure, for example, Pearson correlation coefficient and Chi-square dissimilarity, is another type of similarity or dissimilarity measure methods. The third type of similarity measures is divergence measure, which includes Kullback-Leibler divergence, Jeffrey divergence, Kolmogorov-Smirnov divergence, and Cramer-von Mises divergence. There are some other measures, such as Earth Mover's distance and diffusion distance [14, 15].

The learning to rank model has gained increasing attention in recent years, utilizing machine learning algorithms to optimize the ranking function by tuning some of the parameters and incorporating relevance features [16, 17]. Manifold ranking [18], a graph-based semisupervised learning method, ranks the data through exploiting their intrinsic manifold structure. Manifold ranking was firstly applied to CBIR in [19] and significantly improved image retrieval performance. Liu et al. [20] proposed a graph-based approach for tag ranking, by which a tag graph was built to mine the correlations among tags, and the relevance scores were obtained through a random walk over the similarity graph. These researches demonstrated the effectiveness of graph-based semisupervised learning techniques in solving different ranking problems. However, they are inadequate for the relations in images via pairwise graphs solely. It would be of great benefit to take into consideration the relationship among 3 or more vertices. Such a model capturing higher-order relations is called hypergraph. In a hypergraph, a nonempty set of vertices is defined as a weighted or unweighted hyperedge; the magnitude of the weight represents the degree that the vertices in the hyperedge belong to the same cluster. Agarwal et al. [21] firstly introduced hypergraph to computer vision and proposed a clique averaging graph approximation scheme to solve the clustering problems. Literature [22] formulated the probabilistic interpretation based image-matching problem as the hypergraph convex optimization and achieved a global optimum of the matching criteria. However, they set up three restrictions that are the same degree of all hyperedges, the same number of vertices in two graphs, and a complete match. Sun et al. [23] employed the hypergraph to capture the correlations among different labels for multilabel classification. The proposed hypergraph learning formulation showed the effectiveness on large-scale benchmark datasets, and its approximate least squares formulation maintained efficiency, as well as competitive classification performance. One shortage of their work is that they limited the target applications to linear models and thus did not have a general performance evaluation on other multilabel applications, such as the kernel-induced multilabels. In [24] the spatiotemporal relationship among different patches is captured by the hypergraph structure, and the video object segmentation is modeled as hypergraph partition. Furthermore, weights are added on important hyperedges. The experimental results have shown good segmentation performance on nature scenes. In the case that there are several different types of vertices or hyperedges, the hypergraph is called unified hypergraph. L. Li and T. Li [25] proposed a unified hypergraph model for the personalized news recommendation where users and multiple news entities are involved as different types of vertices, and their

implicit correlations are captured. The recommendation is modeled as a hypergraph ranking problem. The hypergraph learning algorithms have demonstrated their capability of capturing complex high-order relations. Their applications in image retrieval are also promising [3, 4].

3. Ranking on Unified Probabilistic Hypergraph

In this research, we employ a unified probabilistic hypergraph to represent the relations of commercial product images, its textual descriptions, and its categorization labels. We propose a model for searching and ranking images based on hypergraph learning. Conventional visual search systems sort and search images based on the similarity of their visual content. The idea of this model is to learn the relevance of different product features, images visual feature, textual feature, and the hybrid visual-textual feature, and then combine them with the results of visual similarity based retrieval.

3.1. Notation and Problem Definition. Let V represent a finite set of vertices. E represents a family of hyperedges on V , and each hyperedge $e \in E$ contains a list of vertices that belong to V . The hypergraph can be denoted as $G = (V, E, w)$ with a weight function w . The degree of a hyperedge e is defined by $d(e) = |e|$, that is, the number of vertices in e . The degree of a vertex v is defined by $d(v) = \sum_{e \in E} w(e)$, where $w(e)$ is the weight of the hyperedge e . The hypergraph can be formulated to a vertex-hyperedge incidence matrix $H \in R^{|V| \times |E|}$, where each entry $h(v, e)$ is defined as

$$h(v, e) = \begin{cases} 1 & \text{if } v \in e \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Then we have $d(v) = \sum_{e \in E} w(e)h(v, e)$, and $d(e) = \sum_{v \in V} h(v, e)$. Let D_v and D_e denote the diagonal matrices containing the vertex and hyperedge degrees, respectively, and let W be the $|E| \times |E|$ diagonal matrix containing the weights of hyperedges.

Consider a simple example of hypergraph $G = (V, E)$, built as shown in Figure 1. Consider $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}\}$ and $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$. The incidence matrix H is defined as

$$H = \begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ v_2 & 1 & 0 & 0 & 0 & 0 & 0 \\ v_3 & 1 & 1 & 0 & 0 & 0 & 1 \\ v_4 & 0 & 0 & 0 & 0 & 0 & 1 \\ v_5 & 0 & 1 & 0 & 0 & 0 & 0 \\ v_6 & 0 & 1 & 1 & 0 & 0 & 0 \\ v_7 & 0 & 1 & 1 & 0 & 0 & 0 \\ v_8 & 0 & 0 & 1 & 0 & 0 & 0 \\ v_9 & 0 & 0 & 0 & 0 & 1 & 0 \\ v_{10} & 0 & 0 & 0 & 1 & 0 & 0 \\ v_{11} & 0 & 0 & 0 & 1 & 0 & 0 \\ v_{12} & 0 & 0 & 0 & 1 & 0 & 0 \end{array} \quad (2)$$

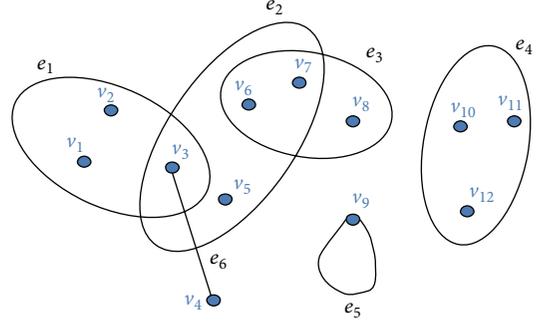


FIGURE 1: An example of a hypergraph.

The problem of ranking on the hypergraph is formulated as follows: given a query vector y , a subset of vertices in the hypergraph $G = (V, E, w)$, a ranking score vector f is produced according to the relevance among vertices in the hypergraph and the query. We define the cost function of f as follows [4]:

$$\begin{aligned} \Omega(f) &= \frac{1}{2} \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} \frac{1}{d(e)} \sum_{\{v_i, v_j\} \in e} w(e) \left\| \frac{f_i}{\sqrt{d(v_i)}} - \frac{f_j}{\sqrt{d(v_j)}} \right\|^2 \\ &+ \mu \sum_{i=1}^{|V|} \|f_i - y_i\|^2, \end{aligned} \quad (3)$$

where $\mu > 0$ is the regulation factor. The first term, known as the normalized hypergraph Laplacian, is a constraint that vertices sharing many incidental hyperedges are supposed to obtain similar ranking scores. The second term is a constraint of the variation between the final ranking score and the initial score.

In order to obtain the optimal solution of the ranking problem, we seek to minimize the cost function:

$$f^* = \arg \min \Omega(f). \quad (4)$$

With the derivations in [4], we can rewrite the cost function as

$$\Omega(f) = f^T (I - \Theta) f + \mu (f - y)^T (f - y), \quad (5)$$

where $\Theta = D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2}$. Then the optimal f^* can be obtained by differentiating $\Omega(f)$ with respect to f :

$$\left. \frac{\partial \Omega}{\partial f} \right|_{f=f^*} = (I - \Theta) f^* + \mu (f - y)^* = 0, \quad (6)$$

$$f^* = \left(I - \frac{1}{1 + \mu} \Theta \right)^{-1} y. \quad (7)$$

3.2. Unified Probabilistic Hypergraph Ranking Model. In the following we will explain our improved hypergraph formulation for the product retrieval and ranking. In a typical

online shopping system there are three different types of information representing a product. They are product image, product name and description, and product labels, which are discussed in detail in Section 4. With these three types of information we design 6 types of hyperedges. Each image in the product image dataset is considered as a vertex in the unified hypergraph. Let X denote the product image pool, and $x_i \in X$ is a particular product image. Let F denote the visual feature description of the images or, say, visual words, let S denote the set of product styles, and let N be the name and description of the product. The unified hypergraph G that contains 6 different types of hyperedge could represent the following implicit relations:

- (1) E^{FSN} (the set of images feature-style-name hyperedges): the products share the same product name, product style, and visual feature word.
- (2) E^{FS} (the set of images feature-style hyperedges): the products, which belong to a certain product style, contain the same visual feature word.
- (3) E^{FN} (the set of images feature-name hyperedges): the products, containing the same visual feature word, share a common keyword in name.
- (4) E^{F} (the set of images visual feature hyperedges): the product images might contain the same visual feature word.
- (5) E^{S} (the set of images style hyperedges): the products belong to the same product style.
- (6) E^{N} (the set of images name hyperedges): the products have similar keywords in their name and description.

Typically we assign 1 to the weights of these hyperedges. Rather than traditional hypergraph structure, in which an image vertex x_i is assigned to a hyperedge e_j in a binary way (i.e., $h(x, e)$ is either 1 or 0), we propose a probabilistic hypergraph to describe the relation between vertex and hyperedge. For hyperedge E^{F} , each image vertex is treated as a centroid, and the hyperedge is formed by the centroid image and its k -nearest neighbors. The incidence matrix H of the probabilistic hypergraph is defined as follows:

$$h(x_i, e_j) = \begin{cases} \text{Sim}(x_j, x_i) & \text{if } x_i \in e_j \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

where x_j is the centroid of e_j . In the proposed formulation, a vertex x_i is softly assigned to a hyperedge e_j based on the similarity between x_j and x_i , which overcomes the limitation of truncation loss with the binary assignment. Besides, we use a parameter to set the desired similarity.

Figure 2 demonstrates an example of constructing such hyperedges. Each vertex and its top 3 similar neighbors form a hyperedge. We set a constraint that only the vertices with a pair similarity larger than 0.4 would be connected

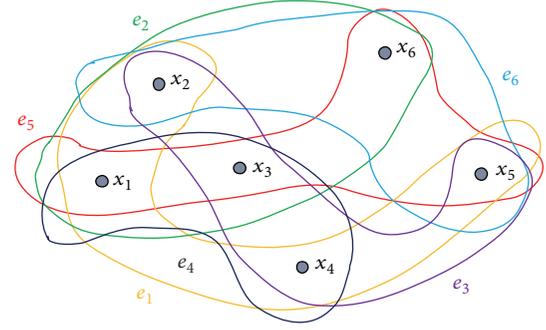


FIGURE 2: An example of probabilistic hypergraph.

into a hyperedge. The incidence matrix H of the proposed probabilistic hypergraph is

$$H = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ x_1 & 1 & 0.81 & 0 & 0.63 & 0.45 & 0 \\ x_2 & 0.81 & 1 & 0.43 & 0.52 & 0 & 0 \\ x_3 & 0 & 0.43 & 1 & 0.83 & 0.62 & 0 \\ x_4 & 0.63 & 0.52 & 0.83 & 1 & 0 & 0 \\ x_5 & 0.45 & 0 & 0.62 & 0 & 1 & 0.8 \\ x_6 & 0 & 0 & 0 & 0 & 0.8 & 1 \end{matrix} \quad (9)$$

With the hyperedges as designed above we can form the 6 types of unified weight matrix W and have the vertex-hyperedge incidence matrix H . The size of both matrices depends on the cardinality of product image dataset involved, and they are all sparse matrices. As a result, the computation of the proposed hypergraph ranking algorithm is fast. It is implemented in two stages: offline training and online ranking. In the offline training stage, we construct the unified hypergraph with matrices H and W derived from above. Then based on the matrices, we calculate the vertex degree matrix D_v and the hyperedge degree matrix D_e . Finally $(I - (1/(1 + \mu))\Theta)^{-1}$ can be computed, where $\Theta = D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2}$. Note that $I - (1/(1 + \mu))\Theta$ is invertible, since the hyperedge E^{kNN} ensures that H is full rank. Then the online ranking procedure can be described as follows: firstly build the query vector y , and secondly compute the ranking score vector f^* . The elements of the preranked relevant images are set to 1, and the others are 0. The procedures are described in Algorithm 1.

Algorithm 1 (the unified probabilistic hypergraph ranking algorithm description).

Input. The inputs are initial query vector of ranked vertices y , similarity matrix Sim , matrix of textual features S , N , and matrix of visual features F for all products vertices.

Output. The output is vector of optimal ranked product vertices f^* :

- (1) Construct the vertex-hyperedge incidence matrix H_1 of hyperedge E^{F} based on (8).

TABLE 1: A typical product representation in a system.

| Product image | Product name and short description | Product labels |
|---|---|---|
|  | (i) BCBGMAXAZRIA Dress, Leyla Sleeveless Keyhole Lace Fit and Flare (ii) Round neck, sleeveless, keyhole front, contrast lace bodice, contrast waistband, drop waist | Occasion: cocktail Type: fit and flare Type: lace Length: short Sleeve length: sleeveless |

- (2) Construct the vertex-hyperedge incidence matrix H_2 of hyperedges $E^S, E^N, E^{FS}, E^{FN},$ and E^{FSN} based on (1).
- (3) Form the incidence matrix H by concatenating H_1 to H_2 .
- (4) Calculate the vertex degree matrix D_v and the hyper-edge degree matrix D_e using $d(e_i) = \sum_{v \in V} h(v_j, e_i)$ and $d(v_i) = \sum_{e \in E} w(e_j)h(v_i, e_j)$, respectively.
- (5) Compute $(I - (1/(1 + \mu))\Theta)^{-1}$, where $\Theta = D_v^{-1/2}HWD_e^{-1}H^T D_v^{-1/2}$.
- (6) Compute the optimal f^* by (7).

4. Experimental Results

In the experiment, we build the unified images hypergraph using different combinations of hyperedges to test the effect of different factors on the ranking performance. We then investigate the performance of different hypergraphs. The superiority of the transductive inference is demonstrated in handling the queries that lack user labels. We use the visual similarity based ranking as a baseline. We compare the different hypergraph-based ranking models with the visual similarity ranking. Also we use the visual similarity ranking score to deduce the preranked score in hypergraph ranking.

For an online shopping system, a product is represented by three types of information, as shown in Table 1: (1) images, which demonstrate the product visually (this usually has several photos taken from different viewpoint); (2) name, which is the name of the product or gives a brief description of the product; (3) labels, which are the textual tags that classify the product into different categories according to the sorting rules. For example, for apparel products, we could have different categories like style, length, sleeve length, occasions, and so forth.

The product image dataset used in the experiment is obtained from a list of prominent brands of women apparel. It contains 3 product categories, 58 brands, and 4210 images. We use different dress categories such as type, length, and sleeve length to form the set of product styles, which contains 7 types, 3 lengths, and 6 sleeve lengths. The product name is the product brand, its style name, and a short description. Here we generate a bag of words to represent it. For visual features, we first extract a color boosted SIFT feature [26], which captures the product color feature and its local patterns, and

then quantize the visual feature descriptors into 65 visual words. For parameters k and μ , we follow the setting in literature [4], where they are empirically set to 100 and 0.001. We choose Normalized Discounted Cumulative Gain (NDCG) [27] to evaluate the ranking performance. NDCG is widely adopted in machine learning approaches to ranking. It is designed for situations of nonbinary notions of relevance. In our research, an experiment participant is asked to judge the relevance of each retrieval result to the query. Each returned image is to be judged on a scale of 0–3 with rel = 0 meaning irrelevant, rel = 3 meaning completely relevant, and rel = 1 and rel = 2 meaning “somewhere in between.” NDCG at position k is defined as

$$\text{NDCG}@k = \frac{\sum_{i=1}^k ((2^{\text{rel}_i} - 1) / \log_2(i + 1))}{\text{IDCG}}. \quad (10)$$

In our experiment, the relevance is assessed over a subset of the collection that is formed from the top k results returned by different ranking methods. Human’s judgments are subjective, idiosyncratic, and variable. Thus, we employ kappa statistic [28] to evaluate the degree of agreement between judges:

$$\text{kappa} = \frac{p(A) - p(E)}{1 - p(E)}, \quad (11)$$

where $p(A)$ is the proportion of the times the judges agreed and $p(E)$ is the proportion of the times they should agree by chance.

In our proposed method, we integrate 7 different relations and hyperedges into constructing product hypergraph so that it effectively represents the product image dataset. The hypergraph also encloses multiple correlations among different visual words and text features. To evaluate the effectiveness of such a representation in product search, we consider different hypergraph constructions with different hyperedge integration. Figure 3 illustrates the ranking performance in terms of average NDCG at different depths of 10, 20, and 30. It is evident that the hybrid hypergraph (FSN, FN, and FS) outperforms the simple construction of hypergraph (F) and the visual similarity based ranking (k NN). And the proposed unified hybrid hypergraph FSN achieves the best performance. The reason for this is quite straightforward: high-order correlations among product visual features and its textual labels are well captured in our unified hypergraph model. The representation and description of a product are extensively enhanced in database.

In Figure 4, an example of a query is demonstrated, in which the system cannot find the best match at the top 10. With the similarity ranking, a black tuxedo jumpsuit is recognized as dress, pants, and coats, while with the proposed unified probabilistic hypergraph learning ranking, the system provides a series of products with similar styles, which is meaningful for the online shoppers. The reason is that we not only capture the visual feature and textual feature separately, but also model the correlations between them. In this way, improved search results are produced.

Then, we use kappa statistics to assess relevance. Here rel = 1 and rel = 2 are considered as agreed. If two judges agree

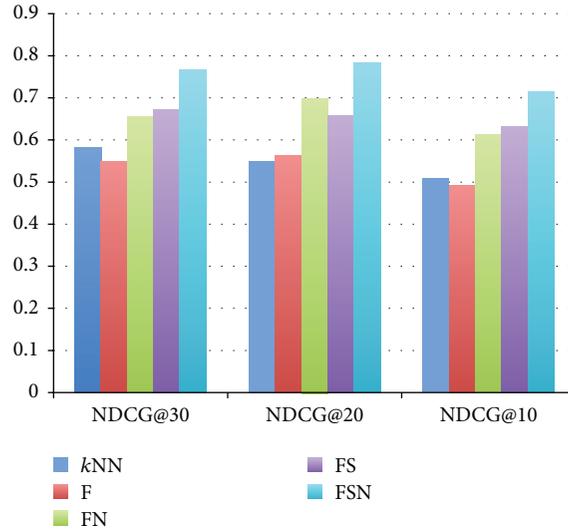


FIGURE 3: Performance comparison of different hypergraph constructions.



FIGURE 4: Top 10 retrieval results with different ranking methods. (a) Visual similarity based ranking. (b) Proposed unified probabilistic hypergraph learning ranking.

on relevance for all results, kappa = 1. If they agree at the probability of agreement by chance, kappa = 0. kappa < 0, if they agree at a rate below random probability. Table 2 gives an example of relevance rating data by two judges on our experiment results. The kappa statistic value between judge 1 and judge 2 is calculated with (11) as 0.7740. The kappa values for all pairs of judges are evaluated in the same way, and their average pairwise kappa value is 0.7483. The level of agreement in our experiments falls in the range of “fair” (0.67–0.8). As a result, the evaluation results of our experiments are proved to be valid.

5. Conclusion

In this paper, we address the problem of ranking in product search by image. We focus on integration of various types of product textual information and visual image. We introduce a hypergraph learning approach to the visual product search and propose a more comprehensive and robust ranking model. In this way the supervised classification and

TABLE 2: An example of relevance rating data by two judges.

| | Judge 1 | | | Total |
|-------------|----------|---------|-------------|-------|
| | Relevant | Between | Nonrelevant | |
| Judge 2 | | | | |
| Relevant | 132 | 11 | 0 | 143 |
| Between | 21 | 207 | 21 | 249 |
| Nonrelevant | 0 | 14 | 74 | 88 |
| Total | 153 | 232 | 95 | 480 |

unsupervised visual search are well balanced. Specifically, we construct the hypergraph by combining three types of product information that embed the relevance among textual features and visual images. Experimental results show that the proposed hypergraph learning framework is a promising ranking scheme for product search. In future work we will consider exploring the adaptive feature weight and other hypergraph learning operators.

Competing Interests

The authors declare that they have no competing interests.

References

- [1] W. H. Hsu, L. S. Kennedy, and S.-F. Chang, "Video search reranking through random walk over document-level context graph," in *Proceedings of the 15th ACM International Conference on Multimedia (MM '07)*, pp. 971–980, September 2007.
- [2] Y. Jing and S. Baluja, "VisualRank: applying pagerank to large-scale image search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1877–1890, 2008.
- [3] Y. Huang, Q. Liu, S. Zhang, and D. N. Metaxas, "Image retrieval via probabilistic hypergraph ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 3376–3383, IEEE, San Francisco, Calif, USA, June 2010.
- [4] Y. Gao, M. Wang, Z.-J. Zha, J. Shen, X. Li, and X. Wu, "Visual-textual joint relevance learning for tag-based social image search," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 363–376, 2013.
- [5] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: from pairwise approach to listwise approach," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pp. 129–136, ACM, Corvallis, Ore, USA, June 2007.
- [6] S. E. Robertson and D. A. Hull, *The TREC-9 Filtering Track Final Report*, 2000.
- [7] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.
- [8] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)*, pp. 275–281, Melbourne, Australia, August 1998.
- [9] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.
- [10] S. Brin and L. Page, "Reprint of: the anatomy of a large-scale hypertextual web search engine," *Computer Networks*, vol. 56, no. 18, pp. 3825–3833, 2012.
- [11] M. R. Henzinger, "Hyperlink analysis for the web," *IEEE Internet Computing*, vol. 5, no. 1, pp. 45–50, 2001.
- [12] R. Baeza-Yates and E. Davis, *Web Page Ranking Using Link Attributes*, ACM, 2004.
- [13] R. Lempel and S. Moran, "The stochastic approach for link-structure analysis (SALSA) and the TKC effect," *Computer Networks*, vol. 33, no. 1, pp. 387–401, 2000.
- [14] M. Arevalillo-Herráez, J. Domingo, and F. J. Ferri, "Combining similarity measures in content-based image retrieval," *Pattern Recognition Letters*, vol. 29, no. 16, pp. 2174–2181, 2008.
- [15] J. Collins and K. Okada, "A comparative study of similarity measures for content-based medical image retrieval," in *Proceedings of the Cross Language Evaluation Forum Conference (CLEF '12)*, Rome, Italy, September 2012.
- [16] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: from pairwise approach to listwise approach," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pp. 129–136, ACM, June 2007.
- [17] T.-Y. Liu, "Learning to rank for Information retrieval," *Foundations and Trends in Information Retrieval*, vol. 3, no. 3, pp. 225–231, 2009.
- [18] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *Advances in Neural Information Processing Systems*, vol. 16, no. 16, pp. 321–328, 2004.
- [19] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang, "Generalized manifold-ranking-based image retrieval," *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 3170–3177, 2006.
- [20] D. Liu, X. S. Hua, L. Yang, M. Wang, and H. J. Zhang, "Tag ranking," in *Proceedings of the ACM 18th International Conference on World Wide Web (WWW '09)*, Madrid, Spain, April 2009.
- [21] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie, "Beyond pairwise clustering," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 838–845, IEEE, San Diego, Calif, USA, June 2005.
- [22] R. Zass and A. Shashua, "Probabilistic graph and hypergraph matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, IEEE, Anchorage, Alaska, USA, June 2008.
- [23] L. Sun, S. Ji, and J. Ye, "Hypergraph spectral learning for multi-label classification," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, pp. 668–676, ACM, 2008.
- [24] Y. Huang, Q. Liu, and D. Metaxas, "Video object segmentation by hypergraph cut," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 1738–1745, June 2009.
- [25] L. Li and T. Li, "News recommendation via hypergraph learning: encapsulation of user behavior and news content," in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM '13)*, pp. 305–314, ACM, Rome, Italy, February 2013.
- [26] K. Zeng, N. Wu, and K. K. Yen, "A color boosted local feature extraction method for mobile product search," *International Journal on Recent Trends in Engineering and Technology*, vol. 10, no. 2, pp. 78–84, 2014.
- [27] G. Karypis and V. Kumar, "Multilevel k-way hypergraph partitioning," *VLSI Design*, vol. 11, no. 3, pp. 285–300, 2000.
- [28] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, New York, NY, USA, 2009.

Research Article

Color, Scale, and Rotation Independent Multiple License Plates Detection in Videos and Still Images

Narasimha Reddy Soora and Parag S. Deshpande

Department of Computer Science & Engineering, Visvesvaraya National Institute of Technology, Nagpur 440010, India

Correspondence should be addressed to Narasimha Reddy Soora; snreddy75@yahoo.co.uk

Received 4 March 2016; Revised 7 June 2016; Accepted 16 June 2016

Academic Editor: Daniel Zaldivar

Copyright © 2016 N. R. Soora and P. S. Deshpande. 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.

Most of the existing license plate (LP) detection systems have shown significant development in the processing of the images, with restrictions related to environmental conditions and plate variations. With increased mobility and internationalization, there is a need to develop a universal LP detection system, which can handle multiple LPs of many countries and any vehicle, in an open environment and all weather conditions, having different plate variations. This paper presents a novel LP detection method using different clustering techniques based on geometrical properties of the LP characters and proposed a new character extraction method, for noisy/missed character components of the LP due to the presence of noise between LP characters and LP border. The proposed method detects multiple LPs from an input image or video, having different plate variations, under different environmental and weather conditions because of the geometrical properties of the set of characters in the LP. The proposed method is tested using standard media-lab and Application Oriented License Plate (AOLP) benchmark LP recognition databases and achieved the success rates of 97.3% and 93.7%, respectively. Results clearly indicate that the proposed approach is comparable to the previously published papers, which evaluated their performance on publicly available benchmark LP databases.

1. Introduction

License plate recognition (LPR) system plays a key role in intelligent transportation systems, such as traffic control, parking lot access control, electronic toll collection, and information management. Typical LPR system contains four processing steps. The first step is to get the image or video from the camera. The second step is LP detection from the input image. The third step is to extract the characters from the LP and the final step is to recognize the extracted characters using different classifiers. These four steps can be achieved by the combination of different techniques of image processing and pattern recognition. Out of these four steps, the LP detection and character recognition steps are very crucial for the success of LPR systems.

LP detection systems have shown significant development for more than a decade with good performance reports, but most of these systems' evaluation is carried on proprietary data sets, having controlled conditions on environment and plate variations. To assess the performance of the LP

detection methods, there is a need for a common publicly available benchmark LP data set, which should contain videos and images taken in an open environment and with different plate variations. A common publicly available benchmark LP data set, for performance evaluation of LPR systems, which is initiated by Anagnostopoulos et al. in paper [1] and contains 741 still images of Greek LPs with several open environmental conditions and different plate variations is present at [2]. For evaluation of the proposed approach, we have used 741 still images of media-lab Greek LP database, 159 Indian, and Israeli LPs from videos and still images. As media-lab Greek LP database missed motorcycles, vehicles with rotated LPs, the combination of different types of vehicles, and more than one motorcycle in a single image, an appropriate care is taken while selecting Indian and Israeli LP images, to achieve all the combinations of plate variations which are missed by media-lab Greek LP database.

In this paper, we have proposed a new approach for finding the LP/LPs in an image using various clustering techniques on geometrical properties of the LP characters, and

a new approach for finding and extracting the noisy/missed characters of the LP/LPs, due to the presence of noise such as dirt or screw or stamp between LP characters and LP border. The clustering techniques proposed in this paper use geometrical properties of the components of LP characters, such as the distance between the components, the angle between the components, and the height of the components, to find the probable LP/LPs. This is the first time that different clustering techniques are applied on geometrical properties of the components of an input image for finding the probable LP/LPs. The proposed geometry-based clustering method for finding the vehicle LP/LPs is scale and rotational invariant and is suitable for many countries LP detection, for any type of vehicles and motorcycles having different plate variations.

The performance of the proposed LP detection method is more prominent when compared with other competitive LP detection methods from the literature, by taking into consideration publicly available benchmark LP databases. It is inappropriate to declare which methods are better because in most of the previously published methods the performance evaluations were carried on proprietary data sets having restricted conditions and were not revealed to the public, to assess their performance. In this paper, we are proposing new methods for LP detection, noisy/missed character extraction, and LP characters rotation correction. New findings in this paper are as follows:

- (i) Proposing a new method for LP detection, using distance-based, line-based, and height-based clustering techniques, on geometrical properties of the LP components.
- (ii) Proposing a new method to remove unwanted clustered components, using the thinning and resizing technique.
- (iii) Proposing a new method for correcting LP rotation of the probable LP cluster components, using the average angle amongst successive probable LP cluster components' left-top coordinates and x -axis.
- (iv) Proposing a new method to extract the noisy/missed characters of the probable LP, because of the presence of noise between LP characters and LP border.

The remaining sections of this paper are planned as follows. Section 2 exhibits the existing similar research. Section 3 describes the proposed approach for multiple LPs detection. Section 4 elaborates on the proposed methodology for multiple LPs detection. Section 5 describes the extraction of noisy/missed characters due to the presence of noise between LP characters and LP border. This section also describes a method for probable LP characters rotation if the vehicle LP is rotated. Experimental results are discussed in Section 6 and Conclusion in Section 7.

2. Existing Similar Research

Many LPR algorithms have been proposed in the literature for the past ten years and even today LP detection remains the challenging area due to different environmental conditions and plate variations. In the literature, there is no LP

detection method which will work for many countries, for all types of vehicles and motorcycles, without any constraint. LP detection is challenging and crucial in LPR systems, which influences the recognition rate. Most of the existing LP detection papers from the literature are based on edge information, morphological operations, template matching, and color information of the LP.

Lee et al. in paper [3] proposed a color image processing (CIP) method to extract the LP of Korean car's, based on LP background and LP characters color using the color histogram. A Neural Network (NN) classifier is used to classify a color. This paper used the aspect ratio of the LP region to select the most probable LP and reported 91.25% success rate for LP detection over 80 car images. The drawback of paper [3] is that the LP detection will not work properly if the vehicle color matches either background color of the LP or color of the LP characters. A hybrid LP localization scheme is presented by Bai and Liu in paper [4], based on the edge statistics and morphology (ESM). The proposed approach had four sections. Section 1 handles the vertical edge detection, Section 2 takes care of the edge statistical analysis, Section 3 finds the hierarchical-based LP location, and Section 4 finds the morphology-based LP extraction. This paper reported 99.6% overall success rate for detecting the LP out of 9825 images. The drawback of paper [4] is that it uses edge information and morphology-based approaches to detect the LP. Some LPs are not so easy to detect using edge information and morphology-based approaches have to define Structuring Element (SE) to perform morphological operations to find the probable LP/LPs from an input image. Defining a particular SE to perform morphology-based operations to detect the probable LP/LPs is a nongeneric approach and will fail to detect the LP/LPs from the input images under various characteristics of LP/LPs in the images.

Yang et al. in paper [5] proposed a new method based on fixed color collocation (FCC) to locate the LP. This method used the color collocation of the plate's background and characters, to recognize the LPs. This paper reported 95% success rate for LP detection. The drawback of paper [5] is also similar to the drawback of paper [3]. Anagnostopoulos et al. in paper [6] proposed a new image segmentation method called sliding concentric windows (SCW) for LP detection. The SCW method works based on local irregularities in the image. The method uses statistics such as standard deviation and mean value, for possible LP location. SCW uses two concentric windows A and B with different sizes, to scan the image from left to right and top to bottom to find the mean and standard deviation of the regions of the concentric windows. If the ratio of the statistical measurements exceeds a threshold value set by the user, then the central pixel of the two concentric windows is considered to be the part of LP. This paper reported a success rate of 96.5% for LP detection using media-lab proprietary LP data set. The limitation of paper [6] is that the statistical measurement threshold value set by the user has to be decided according to the application after a trial-and-error procedure, which is not a generic solution.

Faradji et al. in paper [7] proposed a real-time and robust (RTR) method to find LP location. Finding LP location has

several stages, with the combination of Sobel mask, histogram analysis, and morphological operations. The overall success rate for detecting the LP by this paper was 83.5%. The limitation of paper [7] is the same as that in paper [4]. Huang et al. in paper [8] proposed LPR strategy for motorcycles for checking annual inspection status. The LP data set considered by this paper contains only motorcycles having the LP characters falling in only one line. The method proposed in this paper finds the LP using search window with the help of horizontal and vertical projections (SWHVP) and reported an average LP detection rate of 97.55%. The drawback of paper [8] is that it is not mentioned how to get the initial size of the search window to perform horizontal and vertical projections. It also used morphology-based dilation operation which is not a generic solution to detect the LP/LPs as explained in paper [4] drawback. Wen et al. in paper [9] proposed two methods to find the LP from an input image (two pass). These two methods are based on Connected Component Analysis (CCA) model. Before applying these methods, the input image is binarized using improved Bernsen algorithm, to remove shadows and uneven illumination. Method 1 is used to find the candidate regions based on prior knowledge of the LP. The frame is detected using CCA methodology. If the frame is broken, the LP cannot be detected correctly. When the LP is not detected using Method 1, then Method 2 is adopted. Method 2 extracts the LP using large numeral extraction technique. This paper reported a success rate of 97.16%. The drawback of paper [9] is that the proposed Method 1 fails to detect the LP/LPs from the input image if the LP frame is broken. The drawback of the Method 2 is that it will fail to detect the LP/LPs from the input image if the LP/LPs are not in horizontal position.

Haneda and Hanaizumi in paper [10] proposed RELIP algorithm, which performs a global search for the probable LP using multiple templates, 3D cross-correlation function, and Principal Component Expansion. This paper uses corner detection to remove deformation of LPs. RELIP reported 97% LP detection success rate. The drawback of paper [10] is that it uses spatial similarity with an LP/LPs template to detect the LP which is not a generic solution in the real world context as the LPs having various deformations such as tilt, rotation, and pan from various viewpoints. Zhou et al. in paper [11] proposed a new approach for LP detection based on Principal Visual Word (PVW) discovering and visual word matching. In visual word matching, it will compare the extracted SIFT features of the test image with all discovered PVW and locate the LP based on matching results. This method published 93.2% success rate on the proprietary data set and 84.8% success rate on Caltech dataset. The drawback of paper [11] is that it works based on the prior knowledge of the LP.

Al-Ghaili et al. in paper [12] proposed a new approach in which a color image is converted to grayscale and then the adaptive threshold is applied on the grayscale image to convert it into a binary image. ULEA method is applied to the grayscale image to enhance the quality, by removing the noise. Next, VEDA is applied to detect the LP from the input image. In order to detect the true LP, some statistical and logical operations are applied. The success rate reported by this paper was 91.65% for LP detection. The drawback

of paper [12] is that it extracts the LP/LPs from the input image based on extracting vertical edges, which is the same as the drawback of paper [4]. Hsu et al. in paper [13] proposed a new approach (AOLP) for detecting LP candidates, using Expectation-Maximization clustering method on vertical edges of grayscale images. This paper reported 93.33% success rate on AOLP proprietary benchmark LP data set and reported 92.1% success rate on media-lab benchmark LP data set, for LP detection. It is mentioned in the paper that the LPR solution is designed primarily based on LPs of Taiwan and is not optimal for other countries. The drawback of paper [13] is the same as that in paper [4] because its LP detection is based on extracting vertical edges.

Abo Samra and Khalefah in paper [14] proposed a new LP localization algorithm using dynamic image processing techniques and genetic algorithms (GA) (DIP-GA). CCA technique is used to detect the candidate objects of the input image and improved the CCA technique with the help of modified GA. The system is made adaptable to any country by introducing a scale-invariant geometric relationship matrix to model the LP symbols. The speed of the LP detection is improved by introducing two new crossover operators. The system reported 97.61% success rate using publicly available media-lab benchmark LP database by considering only 335 images out of 741 images. This paper also reported 98.75% success rate using proprietary data set having 800 image samples and reported 98.41% overall accuracy. The drawback of paper [14] is that it is not able to detect multiple LPs in an image.

The LP detection methods which use edge information and morphological operations mainly focus on finding the components which are rectangular in shape with specific aspect ratio. Such type of LP extraction methods will fail to identify the LP if the LP does not follow a rectangular shape with proper aspect ratio. The problem with template matching LP detection methods is that it will not work for all types of plate variations. The color based LP extraction methods use the background color of the LP, to identify the probable LP candidate region because some countries use a particular background color in their LPs. Such type of LPR systems will fail to detect the LP properly if the body color of the vehicle matches with the LP background color. The above categorized LP detection methods used the features of LP in an image, to extract the LP location from an input image. These categories of LP detection methods have limitations in extracting LP/LPs from an image because of the features they adopted to extract LP/LPs.

In this paper, we are proposing for the first time a new LP detection method, which will work for the LP detection of many countries, having any shape, which uses different clustering techniques on geometrical properties of the character components of the LP/LPs in the input image. The advantage of using different clustering techniques on geometrical properties of the character components of the LP is that they are independent of scale, rotation, tilt, and orientation. There are very few techniques/publications in the literature which talk about LP detection methods under various environmental conditions and plate variations

mentioned, which will work for any type of vehicle and motorcycle having any LP shape.

3. Proposed Approach for Multiple License Plates Detection

This paper proposes a new method for LP/LPs detection and noisy characters extraction, for any type of vehicle and motorcycle, having different plate variations, under different environmental and weather conditions. The environmental conditions include different illumination, weather, and background conditions. The plate variations include location of the plate anywhere on the vehicle, many plates in single image, different combination of vehicles with different plate orientations, different sizes of plates, background color of plates, plates with dirt, rotated plates, LPs having two lines of characters and each line of characters are of different size, and tilted LPs. The proposed method can be articulated as a generalized method for identifying the LP/LPs, because it is independent of plate variations under different environmental and weather conditions and can be applicable to many countries and for any vehicle having multiple lines in the LP. In the proposed approach, we have not used any type of edge detection, template matching, morphological operations, and color information of the LPs, which are extensively used by previously published papers, to detect the LP.

The proposed method uses CCA to label the components and applies different clustering techniques on geometrical properties of the labelled components such as the location of the components, the angle between the components, and the height of the components, to extract the probable LP/LPs. In most of the countries, the LP characters are near to each other, are positioned along one or multiple lines, and are similar in height. Based on these properties of LP characters, clustering techniques can be applied on geometrical properties of LP characters to identify the LP/LPs from an input image. Most of these properties are followed by many nations while designing their LPs. That is why the proposed approach can be applicable to detect the LP of many nations which follow the properties of the LP characters mentioned in this paper while designing their LPs. The proposed method contains the following steps.

- (i) Apply preprocessing steps on the input image. If it is a video, convert the video into different frames and then apply preprocessing steps. After completing the preprocessing steps, label the components of the input image to find the number of components N_I and to extract the geometrical properties of the each of N_I components such as left-top coordinates, width, and height.
- (ii) Apply newly proposed distance-based clustering algorithm on each of N_I components' left-top coordinates. This algorithm divides the image components into various distance-based clusters, which are close to each other. Let N_d be the number of clusters formed after distance-based clustering from N_I components of the preprocessed input image.

- (iii) Recluster each N_d distance-based cluster into different line-based clusters. If the components of the distance-based clusters subtend a similar angle between the lines joining left-top coordinates of the components with the x -axis, then construct line-based clusters from such components. Line-based clustering algorithm reclusters N_d distance-based cluster components into N_l line-based clusters, which are in line and close to each other.
- (iv) Recluster each N_l line-based cluster, based on the cluster components height. This is height-based clustering, which reclusters N_l line-based clusters into N_h height-based clusters.
- (v) After the distance, line, and height-based clustering techniques, the resultant clustered components have the properties such as near to each other, positioned along a line, and being similar in height. These properties belong to the characters of LP of any vehicle and motorcycle in the world.
- (vi) In the resultant N_h clusters, there may be a few non-LP clusters in which all the components follow the properties of LP characters. In order to remove such non-LP clusters, apply the thinning and resizing technique on N_h height-based clusters. Let N_{tr} be the number of probable LP clusters in the image, after applying thinning and resizing technique.
- (vii) Refine N_{tr} clusters further by finding the border of the cluster components. If the border percentage for each cluster is less than predefined threshold, remove such clusters from the list of probable LP cluster (N_{tr}). After this step, let N be the number of probable LP clusters.
- (viii) Now, apply the newly proposed character extraction method to extract noisy/missed characters of the LP, on each of " N " probable LPs, due to the presence of noise such as screw or dirt or stamps between LP characters and border of the LP.
- (ix) After extracting the noisy/missed characters, rotate the probable LP characters using the average angle between the lines joining adjacent components' left-top coordinates and the x -axis, so that all the probable LP characters will be horizontal to the x -axis.

The above mentioned outlined procedures are explained in detail in the following sections.

4. Detailed Description of the Proposed Approach for Multiple License Plates Detection from Videos and Still Images

This section describes in detail the proposed approach to find the probable LP/LPs in an image with the help of various clustering, thinning, and resizing techniques. This section also explains the method and the need to further refine the probable LP/LPs by finding the border of the LP/LPs.

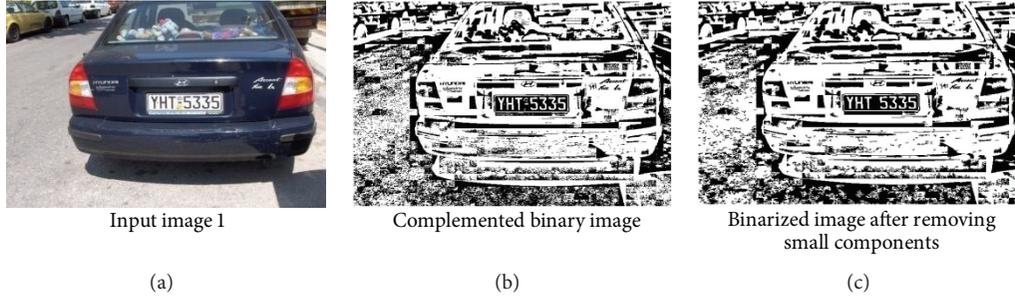


FIGURE 1: Results of preprocessing stage.

4.1. Preprocessing Stage

Steps

- (1) Due to the effect of illumination in an open environment and the presence of shadows, it is very difficult to process an input image with the help of traditional threshold binarization methods and will not give satisfactory results. In this paper, we are using Bernsen algorithm to overcome the illumination and shadow problems in an image. Let $f(x, y)$ denote gray value at a point (x, y) of an image. Let (x, y) be the centre of a block of size $(2w+1) \times (2w+1)$ in the image, where w is the number of pixels. Threshold $T(x, y)$ at the point (x, y) can be computed using

$$T(x, y) = \frac{(\max_{-w \leq k, l \leq w} f(x+l, y+k) + \min_{-w \leq k, l \leq w} f(x+l, y+k))}{2}. \quad (1)$$

- (2) Convert the input image 1 (shown in Figure 1) into grayscale. If it is video, convert it into frames and then convert each frame into grayscale. Apply the Bernsen algorithm to overcome from uneven illuminations or shadows present in the grayscale image. Complement the binarized image, whose output is shown in Figure 1 second column. Remove the components, whose height is less than three pixels from the complemented binary image, because no LP character is less than three pixels in height. The image in Figure 1, third column, shows the output after removing the components that are less than three pixels in height.
- (3) To implement the rest of the operations on individual components, such distance-based clustering, line-based clustering, and height-based clustering extract the geometrical properties of individual components such as left-top coordinates, width, and height.
- (4) Geometrical properties of the individual components described in Step 3 can be extracted using the following procedure:

- (a) To find the number of components present in an image, use CCA method to apply labelling to preprocessed image. Let N_I be the number of components present in the input image at this stage.

- (b) After labelling, find left-top coordinates, width, and height of N_I components.
- (c) Crop each individual component from the preprocessed image using left-top coordinates, width, and height.
- (d) Save the cropped components of the input image as separate image components.

4.2. Clustering Stage. The purpose of the clustering stage is to prepare the probable LP clusters from N_I components of the preprocessed input image. In this stage, the proposed system performs three types of clustering techniques, one after the other. The first clustering technique is the distance-based clustering, whose purpose is to divide N_I components of the preprocessed input image into groups which are near to each other. The second clustering technique is the line-based clustering, whose significance is to divide each distance-based cluster into an array of line-based clusters. The components of the line-based clusters are in line and close to each other from the viewpoint of left-top coordinates of the components. The third clustering technique is the height-based clustering, whose significance is to regroup the line-based cluster components which are similar in their height. After all these clustering techniques, the resultant cluster components in each cluster are close to each other, positioned in a line and alike in their heights which are the probable LP/LPs of the input image.

Context dependent variables are cluster_size and max_distance. The first context dependent variable cluster_size indicates the minimum number of characters in a row of the LP and can be tuned to satisfy country specific LP constraints. In our experiments, we have considered cluster_size as 4. The next context dependent variable max_distance is used during various clustering stages which explain the maximum distance that is allowed between the successive components of the LP clusters. The value of the variable max_distance is computed as one-third of the columns of the input image. The proposed geometry-based clustering method has following steps.

Steps

- (1) After preprocessing stage, cluster all the components of the image, based on the distance between left-top coordinates of each individual component,

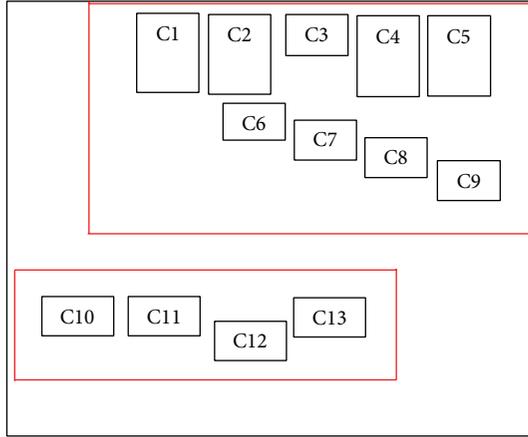


FIGURE 2: An example image showing the components after distance-based clustering.

using distance-based clustering algorithm which is explained in detail in the following steps.

- (2) Distance-based clustering algorithm prepares a matrix (distance matrix) of size $[N_I, N_I]$, where N_I indicates the number of components of the input image after the preprocessing stage. In the distance matrix, the 1st row indicates the distance between the 1st label component and rest of the components' left-top coordinates, 2nd row indicates the distance between 2nd label component and rest of the components' left-top coordinates, and so on. Distance-based clustering algorithm prepares maximum N_I clusters, one for each component with the help of distance matrix.
- (3) Remove those distance-based clusters whose size is less than cluster_size and which are the subset of other distance-based clusters. Retain those distance-based clusters with a large number of components in it when performing the subset removal operation. At this stage, the N_I components of the image are clustered into groups, based on the distance between the left-top coordinates of each individual component.
- (4) Figure 2 shows an example image with the components $C1 \dots C13$, which are clustered into two distance-based clusters highlighted with rectangular border (in red color). The components $C1 \dots C9$ formed as first distance-based cluster and $C10 \dots C13$ as second distance-based cluster.
- (5) Figure 3 shows the resultant image after distance-based clustering. Let N_d be the number of distance-based clusters at this stage, which are shown as ellipses in Figure 3. As most of the components of the input image are very small in size, it is not possible to observe with the naked eye the components formed as distance-based clusters from the input image. Hence, the same is explained by taking an example as shown in Figure 2.



FIGURE 3: Resultant image after distance-based clustering.

- (6) Now, apply line-based clustering technique on each of N_d distance-based clusters, to find those components which are in line with each other.
- (7) In the line-based clustering, consider individual distance-based cluster and take each component from the cluster and draw a line from left-top coordinates (x_1, y_1) of one component to the left-top coordinates (x_2, y_2) of the next component in the current cluster. Find the angle θ between x -axis and the line that is drawn between two left-top coordinates of components as shown in (2). In the same way, find the angle between rest of the components and x -axis, as a matrix (angle matrix) of size $[N, N]$, where N indicates the number of individual components in each distance-based cluster:

$$\theta = \left(\frac{180}{\pi} \right) * \tan^{-1} \left(\frac{(y_2 - y_1)}{(x_2 - x_1)} \right). \quad (2)$$
- (8) Cluster those components as line-based clusters, which subtend similar angle with the x -axis and which are close to each other (based on max_distance), for each row of the angle matrix. Now, remove those line-based clusters, which are less than cluster_size and which are subsets to other line-based clusters. This is the line-based clustering technique.
- (9) Line-based clustering is used to recluster the distance-based cluster components, based on the property of having similar angle and closeness of the cluster components. After this stage, the components of an input image are clustered, using distance-based clustering and line-based clustering techniques.
- (10) Figure 4 shows an example image in which the distance-based clusters (shown in the red border) are divided into line-based clusters (shown in the green border). In Figure 4, the first distance-based cluster with the components $C1 \dots C9$ is divided into two line-based clusters with the components $C1 \dots C5$ and $C6 \dots C9$, indicated with rectangular (green in color) boxes. The second distance-based cluster with the components $C10 \dots C13$ resulted into a line-based cluster with the components $C10, C11,$ and $C13$ and the component $C12$ is removed from the resultant list, because $C12$ is not in line with other components. The resultant image after line-based clustering is shown in

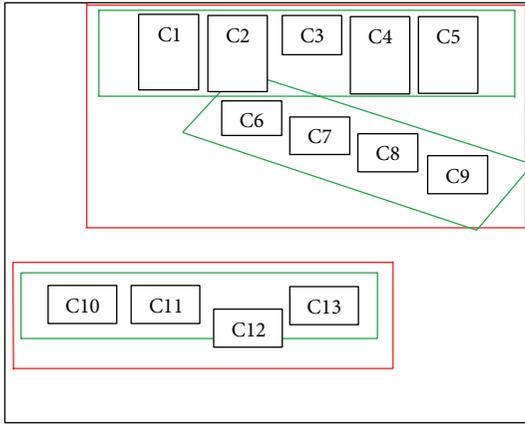


FIGURE 4: An example image showing how the distance-based clusters are divided into the line-based clusters.

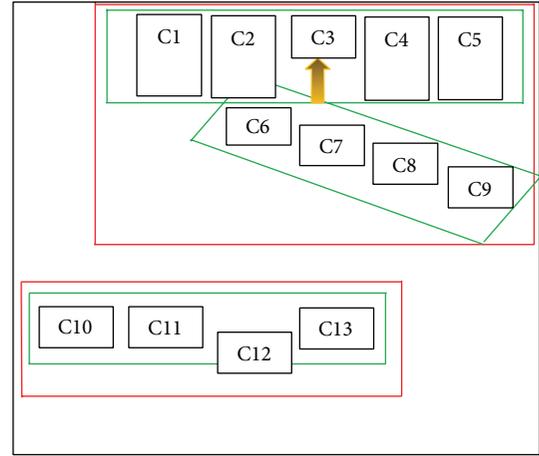


FIGURE 6: An example image showing the height-based clustering.



FIGURE 5: Resultant image after line-based clustering.



FIGURE 7: Resultant image after height-based clustering.

Figure 5 in which the line-based clusters are marked in rectangular boxes.

- (11) There is no difference between Figures 3 and 5 based on the number of components and positions of the components are concerned. The resultant distance-based clusters are shown in the ellipse shape with various colors in Figure 3. The resultant line-based clusters are shown in the rectangular shape with various colors in Figure 5.
- (12) After line-based clustering, apply height-based clustering technique, to remove few unwanted/junk components, which are very close to and in line with the components, but shows much difference in height as compared with other components in each line-based cluster. Now, remove those clusters which are less than cluster_size and which are subsets to other clusters. This is height-based clustering.
- (13) Figure 6 shows an example image in which the component C3 (indicated with the arrow) is showing much difference in height as compared to the rest of the components; it will be removed from the line-based cluster and will result in height-based cluster with the components C1, C2, C4, and C5. Figure 7 shows the image after height-based clustering and the resultant height-based clusters are shown in ellipses.

At this stage, let N_h be the number of height-based clusters.

- (14) After height-based clustering, few of the clusters from the final cluster list may contain all unwanted components, which obey all the characteristics based on distance, line, and height-based clustering. Such junk component clusters can be removed by thinning and resizing technique.
- (15) Apply infinite thinning and resizing technique on each N_h height-based cluster of the image. Thinning is a morphological operation used for skeletonization of the binary image components. When we apply thinning and resizing operations, junk components will retain its shape, but the LP characters will fade away completely. Remove those clusters from the cluster list, which retain its shape after thinning and resizing operations. This technique removes all junk cluster components from the final cluster list. Let N_1 be the number of clusters after removing junk clusters using the thinning and resizing technique. Figure 8 shows image after thinning and resizing technique and the resultant clusters are shown in ellipses shape.

4.3. Finding Border of the License Plate. Contrasting to a typical LPR system, the proposed system first finds the probable LP characters and then finds the border of



FIGURE 8: Resultant image after thinning and resizing technique.



FIGURE 9: Resultant image after license plate border identification.

the components. The reason for finding the border of LP is that there can be a group of non-LP characters in the image with similar properties of LP characters without the border. In order to avoid such type of characters, the system proposes to find the border of the LP.

Predefined Border Percentage: Beta. Beta is a user defined variable, which decides the border percentage that an LP can have. After rigorous experimentations with the help of many datasets from multiple countries, we have come to a conclusion to decide the value of Beta as 70%.

Steps

- (1) Let N_1 be the number of probable LP clusters at this stage. Take each individual component from the cluster and traverse from left-top coordinates of each individual component towards the upward direction, till the traversal reaches the border point or three times the height of each individual character component.
- (2) Apply the same procedure towards the downward direction, to find the bottom border point for each individual component.
- (3) Save the top and bottom border points for all the components of each cluster.
- (4) Find the border percentage for each cluster using the top and bottom border points. Retain a cluster only when the percentage is greater than or equal to the predefined border percentage "Beta." Retained clusters indicate the LP/LPs of an image and its components indicate the individual characters of the LP.
- (5) Remove those clusters which fall below predefined border percentage "Beta." This is another way to further refine the cluster list to get the required LP region. Figure 9 shows the image after LP border identification.

5. Noisy Characters Extraction and License Plate Characters Rotation

Noisy/missed characters extraction stage is to extract few of the LP characters, which may be missed during the previous

stages, because of the presence of noise such as screw or dirt or stamps between the LP border and LP characters. This section proposes a new approach to extract such type of noisy/missed characters. The proposed noisy/missed characters extraction algorithm can be applied after Section 4.3.

Steps

- (1) The proposed algorithm uses (3) to find the noisy LP characters' left-top coordinates (x_2, y_2) at a distance d from the leftmost cluster component's left-top coordinates (x_1, y_1) and the average slope m amongst the cluster components:

$$\begin{aligned} x_2 &= x_1 \pm \left(\frac{d}{\sqrt{1+m^2}} \right), \\ y_2 &= y_1 \pm \left(\frac{m * d}{\sqrt{1+m^2}} \right). \end{aligned} \quad (3)$$

- (2) At this stage, we have few probable LP clusters. Take each probable LP cluster and find the average height and average slope (m) amongst the subsequent cluster components' left-top coordinates and the x -axis.
- (3) From each probable LP cluster, take left-top coordinates of the first component (x_1, y_1) and move in the downward direction to $(1/4)$ th of the average height of probable LP cluster. Now, traverse the image right side, one pixel at a time, using (3) mentioned above, to find any noisy/missed LP character components.
- (4) If any noisy/missed LP character component is found, which is not part of the probable LP cluster component and is not a background pixel, then find the left-top coordinates (x_2, y_2) and the width of the noisy/missed component. To find the left-top coordinates (x_2, y_2) and the width of the noisy/missed component, we have to perform three traversals as described as follows:

- (i) The first traversal is towards the top side of the noisy/missed character component to compensate the left-top coordinate's slope with the average slope. The second traversal is to traverse towards the left side of the missed component in the direction of the average slope, till it reaches

leftmost pixel of the noisy/missed component, within the average height, using (3). After reaching the left most side of the noisy/missed component, fix it as left-top coordinates (x_2, y_2) of the noisy/missed component.

- (ii) Take the newly found left-top coordinates of the noisy/missed LP character component and move towards the right side (third traversal) in the direction of the average slope, till it reaches the right most point of the missed component, within the average height, using (3). The difference between the x -coordinates of the newly found left-top coordinate point (x_1, y_1) and the right most point is the width of the noisy/missed component.
 - (iii) Crop the noisy/missed component using left-top coordinates, width, and the average height of the cluster components.
- (5) Repeat the same procedure, till the traversal reaches the last component of the probable LP cluster component. Repeat the same procedure, for all probable LP clusters.
- (6) Rotate each individual component of the probable LP cluster by using average angle θ , which can be calculated from the average slope (m) amongst the probable LP cluster components using (4). Consider each component of the binary image as $F = \{F(i, j), i = 1, 2, \dots, I, j = 1, 2, \dots, J\}$ and can be defined as shown in (5). One has

$$\theta = \left(\frac{180}{\pi} \right) * \tan^{-1}(m), \quad (4)$$

$$F(i, j) = \begin{cases} 1, & \text{white pixel,} \\ 0, & \text{black pixel.} \end{cases} \quad (5)$$

- (7) Let $F(x, y)$ be the image component before rotation and let $F(x', y')$ be the image component after rotation. Use average angle θ to rotate $F(x, y)$. The equation for each individual pixel of $F(x', y')$ can be obtained by using

$$\begin{aligned} x' &= x * \cos(\theta) + y * \sin(\theta), \\ y' &= -x * \sin(\theta) + y * \cos(\theta). \end{aligned} \quad (6)$$

- (8) In a rotated image, if the average angle of the probable cluster components is above a certain threshold, then there is a chance that the other part of the character component will be present in the target component. In such a case, retain the bigger component from the target component and remove the rest of the components.

6. Experimental Results

The above described concepts are implemented using MATLAB on Intel core i3 processor machine, having 4 GB RAM.

The performance of the proposed LP detection method is compared with some of the competitive LP detection methods, by taking into consideration publicly available media-lab benchmark LP database, Israeli LP images from the web, and proprietary Indian LPs, having different plate variations and weather conditions in an open environment. Total images from all these data sets are 900. To further assess the performance of the proposed LP detection method, we have considered AOLP benchmark LP database having 2049 images with three subsets. For experimentation, we have considered Indian, Israeli, and media-lab Greek benchmark LPs as single data set having different characteristics of the LPs in images as described in Table 1.

In an open environment, there are many possible ways by which we can capture an image from cameras. The proposed geometry-based clustering techniques in this paper are invariant to size, tilt, pan, and rotation. That is why the proposed approach works properly with extreme observation views. There is no restriction on the size of LP characters to detect the LP, which can be observed from Figures 10(a23) and 10(b23). There are very few methods in the literature, which talks about the LP detection of motorcycles, where LP characters fall in two lines and each line of characters are of different size. The proposed method works for any type of vehicles, motorcycles, vans, and trucks having multiple lines of LP characters and also each line of characters having different sizes. The proposed approach will fail to detect the LP/LPs from an input image if the LP characters touch the border of the LP or there are less than cluster_size characters in the LP or there are no characters present in LP at all. Table 1 shows the summarization of the various characteristics of LPs in the images described in Figure 10. For example, Figure 10(a1) (S. number 1 in Table 1) indicates an LP in an input image from Greek in an open environment. In all images, we have assumed neither fixed number of characters in the LP nor the number of lines. We assumed that the sizes of the LP characters may be different due to different view conditions.

Figure 10 shows the sample results for all categories of the LPs, which include images of different plate variations, environmental, and weather conditions from media-lab, Israeli, and proprietary Indian LP databases whose summary is given in Table 1. The odd column of Figure 10 shows the actual image and the even column shows the binarized image. Red border in the binary image of Figure 10 indicates the identification of the LP and without a red border belongs to a failed case. Most of the vehicles in many countries including motorcycles will have only a single line of characters in their LPs, but there may be a chance that few countries like India will have LP characters that will fall into more than one line and there may be a chance that the characters in each line of LP may vary in size as shown in Figures 10(a8), 10(a18), and 10(a19). From such type of the LPs, most of the existing LPR systems detect only the line of characters, which are bigger in size. Hence, such type of LP detection systems will not satisfy the real-time requirements. The proposed approach will work in all such conditions.

Figures 10(a21), 10(b21), 10(a22), and 10(b22) show images with failed LP detection, due to overlapping of LP characters

TABLE 1: Summarization of the characteristics of LPs in the images described in Figure 10.

| S. number | Figure number | Country | Characteristics of the image | Recognized by the proposed approach (yes/no) |
|-----------|----------------|----------------------|---|--|
| 1 | Figure 10(a1) | Greek | Image in open environment | Yes |
| 2 | Figure 10(a2) | Greek | Image with blur | Yes |
| 3 | Figure 10(a3) | Greek | Image with shadow | Yes |
| 4 | Figure 10(a4) | Greek | Image with reflectance and illumination | Yes |
| 5 | Figure 10(a5) | Greek | Image with dirt and screw in the LP | Yes |
| 6 | Figure 10(a6) | Greek | Image with distorted LP | Yes |
| 7 | Figure 10(a7) | Greek | Image with LP at a different place | Yes |
| 8 | Figure 10(a8) | Greek | Image taken at night | Yes |
| 9 | Figure 10(a9) | Greek | Image with dirt and shadows taken on difficult tracks | Yes |
| 10 | Figure 10(a10) | Israel | Image with different tilt | Yes |
| 11 | Figure 10(a11) | Israel | Image with different tilt | Yes |
| 12 | Figure 10(a12) | Greek | Image with an extreme pan | Yes |
| 13 | Figure 10(a13) | Indian | Image with multiple LPs (two cars) | Yes |
| 14 | Figure 10(a14) | Indian | Image with multiple LPs (three motorcycles) | Yes |
| 15 | Figure 10(a15) | Indian | Image with multiple LPs and different combination of vehicles | Yes |
| 16 | Figure 10(a16) | Indian | Image with rotation (an image containing car with left diagonally rotated LP) | Yes |
| 17 | Figure 10(a17) | Indian | Image with rotation (an image containing car with right diagonally rotated LP) | Yes |
| 18 | Figure 10(a18) | Indian | Image with rotation (an image containing motorcycle with left diagonally rotated LP) | Yes |
| 19 | Figure 10(a19) | Indian | Image with rotation (an image containing motorcycle with right diagonally rotated LP) | Yes |
| 20 | Figure 10(a20) | Urdu lingual country | Image from different country | Yes |
| 21 | Figure 10(a21) | Greek | Image with lot of dirt in the LP | No |
| 22 | Figure 10(a22) | Greek | Image with lot of dirt in the LP | No |
| 23 | Figure 10(a23) | Israel | Image with close view of LP | Yes |
| 24 | Figure 10(a24) | Greek | Image with few noisy characters in the LP | Yes |

with LP border because of the presence of lot of dirt between LP characters and LP border. Figures 10(a24) and 10(b24) show images with LPs 2nd (character “K”) and 6th (character “7”) characters touching the border of the LP due to the presence of noise such as screw or dirt or stamp between LP characters and border of the LP. These characters are extracted successfully using the proposed noisy/missed characters extraction algorithm from Section 5. Figures 10(a1)–10(a9), 10(a12), 10(a21), 10(a22), and 10(a24) show Greek LPs, Figures 10(a13)–10(a19) show Indian LPs, Figures 10(a10), 10(a11), and 10(a23) show Israeli LPs, and Figure 10(a20) shows an LP from Urdu lingual country.

From Table 1 and Figure 10 we can conclude that the images are considered from four different countries and we have also considered AOLP benchmark database for performance evaluation which contains images from Taiwan country. With these results, we can claim that the proposed approach will work successfully to identify LP/LPs from an image, whose individual character properties are near to each other, in line with each other, and similar in height in a particular line. There is no restriction on the number of lines

present in the LP of the vehicle and the characters in each line of the LP can be of different size.

The performance comparison amongst few of the prominent LP detection methods and the proposed approach is shown in Table 2. The method proposed by Bai and Liu in paper [4] is superior to the proposed approach with a remarkable LP detection success rate of 99.6%, which supersedes all other methods. The methods proposed by Huang et al. in paper [8], Wen et al. in paper [9], Yang et al. in paper [5], Haneda and Hanaizumi in paper [10], and Anagnostopoulos et al. in paper [6] reported 97.55%, 97.16%, 95.3%, 97%, and 96.5% LP detection rates, respectively, which are less than the proposed method’s success rate of 98.74%. Lee et al. in paper [3] and Al-Ghaili et al. in paper [12] reported 91.25% and 91.65% success rates, but their data set contains only cars. Faradji et al. in paper [7] reported lower success rate, as compared to others. Zhou et al. in paper [11] reported 93.2% success rate on the proprietary data set and 84.8% on Caltech data set. Abo Samra and Khalefah in paper [14] reported 98.75% success rate which is equivalent to the proposed methods success rate of 98.74%. The success rates

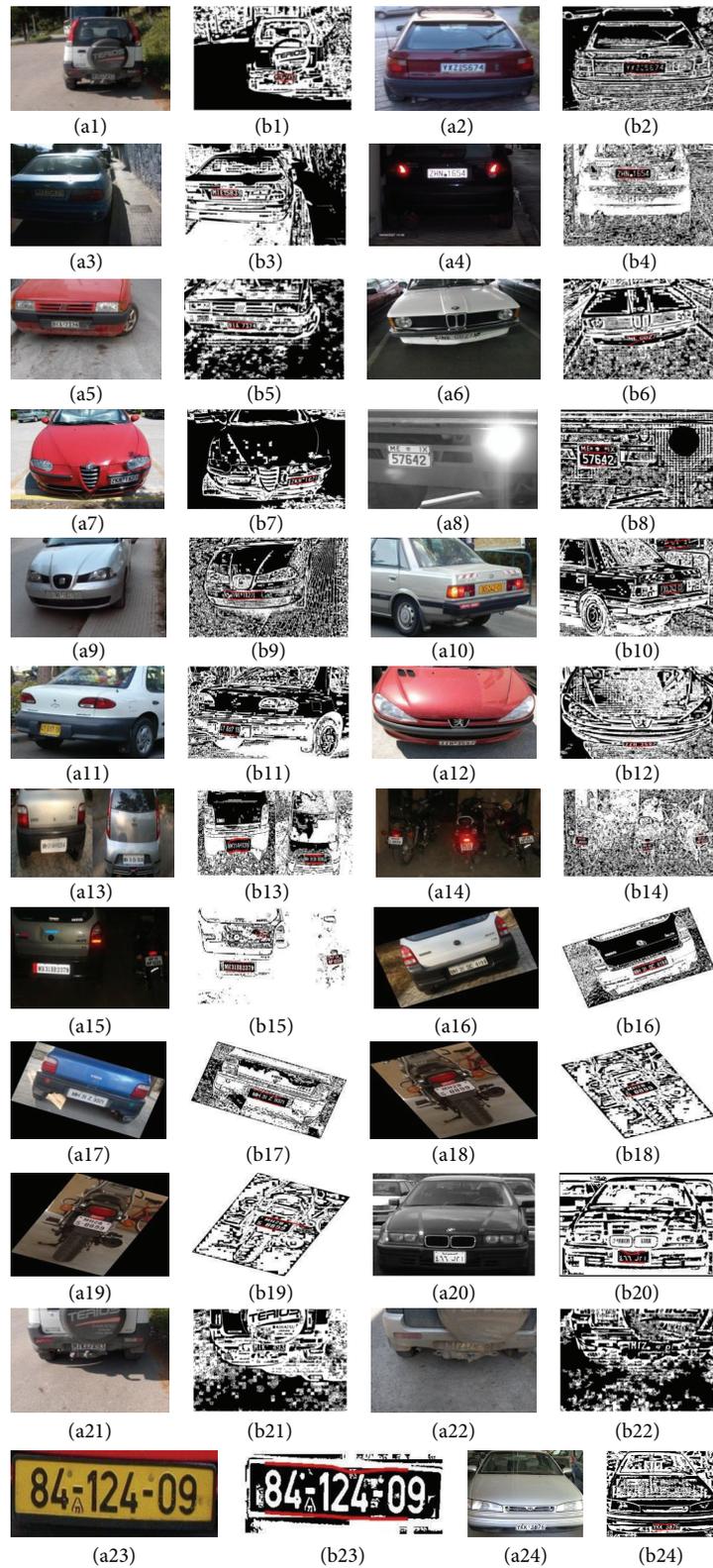


FIGURE 10: Examples of images with all categories of LPs showing LPs detection.

TABLE 2: Performance comparison of the proposed LP detection method with few of the competitive LP detection methods from the literature.

| Algorithm and references | Data set | Number of images in data set | Types of vehicles present in data set | LP detection rate |
|---|---------------------------|------------------------------|---------------------------------------|-------------------|
| CIP [3] | Proprietary | 80 | Cars | 91.25% |
| ESM [4] | Proprietary | 9825 | Vans, trucks, cars | 99.6% |
| FCC [5] | Proprietary | 150 | Not reported | 95.3% |
| SCW [6] | Media-lab (proprietary) | 1334 | Vans, trucks, cars | 96.5% |
| RTR [7] | Proprietary | 400 | Not reported | 83.5% |
| RTR [7] | Caltech | 1999 | Cars | 84.8% |
| SWHVP [8] | Proprietary | 522 | Motorcycles | 97.55% |
| Two pass [9] | Proprietary | 9026 | Cars, trucks | 97.16% |
| RELIP [10] | Proprietary | 100 | Not reported | 97% |
| PVW [11] | Proprietary | 410 | Not reported | 93.2% |
| VEDA [12] | Proprietary | 664 | Cars | 91.65% |
| AOLP [13] | Media-lab | 741 | Vans, trucks, cars | 92.1% |
| AOLP [13] | AOLP (proprietary) | 2049 | Cars and vans | 93.33% |
| DIP-GA [14] | Proprietary | 800 | Vans, trucks, cars | 98.75% |
| DIP-GA [14] | Media-lab | 335 | Vans, trucks, cars | 97.61% |
| Proposed geometry-based clustering method | Media-lab | 741 | Vans, trucks, cars | 97.3% |
| Proposed geometry-based clustering method | Proprietary | 159 | Cars, trucks, motorcycles | 98.74% |
| Proposed geometry-based clustering method | Media-lab and proprietary | 900 | Vans, trucks, cars, motorcycles | 97.56% |
| Proposed geometry-based clustering method | AOLP | 2049 | Cars and vans | 93.7% |

TABLE 3: Performance comparison of the proposed method with SCW and AOLP methods using media-lab and AOLP benchmark LP databases.

| Benchmark LP database | LP database details | Conditions | SCW method | AOLP method | Proposed geometry-based clustering method |
|---------------------------------|---|--|------------|-------------|---|
| Media-lab benchmark LP database | 741 images having vans, trucks, and cars | Open environment, different plate variations | 96.5% | 92.1% | 97.3% |
| AOLP benchmark LP database | 2049 images having vans, trucks, and cars | Open environment, different plate variations | 81.67% | 93.33% | 93.7% |
| Average of success rates | | | 89.09% | 92.72% | 94.66% |

of the above mentioned LP detection systems are based on proprietary data sets.

It is impractical to compare the performances of different LP detection systems which evaluated their performances using proprietary LP data sets. There should be a common, true benchmark LP database, openly available to assess the performance of the proposed LP detection systems. A common, publicly available media-lab benchmark LP database, for the research community, is initiated by Anagnostopoulos et al. in paper [1], which contains Greek vehicle LP images. As the media-lab benchmark LP database is not satisfying all plate variations mentioned in this paper, we coupled the images of Israeli and Indian LPs having cars, vans, trucks, and motorcycles, with media-lab benchmark LP database, to attain all plate variations mentioned. Table 3 shows the performance comparison between SCW method [6], AOLP

method [13], and the proposed approach using media-lab and AOLP benchmark LP databases. Using media-lab benchmark LP database, the proposed method's success rate of 97.3% is better when compared to SCW method's success rate of 96.5% (number of images taken by SCW method is 1334) and is more than AOLP method's success rate of 92.1%. Using AOLP benchmark LP database, the success rate of the proposed approach is 93.7%, which is close to the success rate of 93.33% of the AOLP approach and better than 81.67%, which is the success rate of SCW method. The average success rate of the proposed approach which is based on both the benchmark LP databases is 94.66% and is a bit more than AOLP's average success rate of 92.72% and is better than SCW's average success rate of 89.09% as shown in Table 3. Abo Samra and Khalefah in paper [14] also tested their proposed methods performance using media-lab benchmark

LP database and reported a success rate of 97.61% using 335 images only (instead of 741 images), whereas the proposed method used 741 images from media-lab benchmark LP data set and reported 97.3% success rate which is almost equivalent to the success rate reported in paper [14].

The success rate of the proposed approach using media-lab benchmark LP data set and the proprietary LP data set is 97.56%, which is a bit more as compared to the success rate of 96.5% of the SCW approach. The success rate of SCW method is reported using 1334 images, whereas they made available only 741 images online, as media-lab benchmark LP database. We do not have clarification on rest of (1334 – 741) 593 images. During our experimentation, we observed around 2% of the media-lab LP benchmark database having noisy/missed characters and we have achieved 100% noisy characters extraction from the input images using the proposed noisy/missed characters extraction method.

We observed that most of the LP detection papers from the literature vastly used edge information, template information, morphological operations, and color information of the LPs. These types of methodologies have restrictions when detecting the LP/LPs from the input images as explained in Section 2. In order to overcome the shortcomings of the LP detection methods from the literature which are enlightened in this paper, we have proposed geometry-based clustering techniques which are invariant to color, scale, rotation, and scale variances of the LPs and also proved from Figure 10 that the proposed multiple LPs detection method successfully detects the LPs from the input images taken from the open environment, all weather conditions, and all plate variations mentioned in this paper. Hence, the proposed method has the ability to detect multiple LPs from an input image which follow the properties of the proposed geometry-based clustering techniques.

7. Conclusion

In this paper, we have proposed a new method for LP/LPs detection and noisy/missed characters extraction due to the presence of noise between LP characters and LP border. The proposed method's performance is evaluated on media-lab and AOLP benchmark LP data sets and reported success rates of 97.3% and 93.7%, respectively, which are shown in performance comparison Table 2. The average success rate of the proposed approach (94.66%) is more as compared to SCW (89.09%) and AOLP (92.72%) approaches using both benchmark LP data sets which are shown in Table 3. The proposed approach can detect multiple LPs in an image and is not specific to any country; there is no restriction on the number of characters present in the LPs, the number of lines present in the LPs, and the size of the characters in each line of the LP. As demonstrated in the results, the proposed approach is less restrictive as compared with most of the previously published work and it works for many countries having different plate variations, under different environmental and weather conditions. The proposed approach fails to identify the LP/LPs, if the LP characters are missed, due to

the presence of noise such as extremely dirty or blur or characters of LP touches the LP border.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

References

- [1] C.-N. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas, "License plate recognition from still images and video sequences: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 377–391, 2008.
- [2] I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, E. Kayafas, and C.-N. E. Anagnostopoulos, "Medialab LPR Database," Multimedia Technology Laboratory, National Technical University of Athens, <http://www.medialab.ntua.gr/research/LPRdatabase.html>.
- [3] E. R. Lee, P. K. Kim, and H. J. Kim, "Automatic recognition of a car license plate using color image processing," in *Proceedings of the 1st IEEE International Conference on Image Processing (ICIP '94)*, vol. 2, pp. 301–305, Austin, Tex, USA, 1994.
- [4] H. Bai and C. Liu, "A hybrid license plate extraction method based on edge statistics and morphology," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, vol. 2, pp. 831–834, Cambridge, UK, August 2004.
- [5] Y.-Q. Yang, J. Bai, R.-L. Tian, and N. A. Liu, "A vehicle license plate recognition system based on fixed color collocation," in *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC '05)*, pp. 5394–5397, August 2005.
- [6] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas, "A license plate-recognition algorithm for intelligent transportation system applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 3, pp. 377–391, 2006.
- [7] F. Faradji, A. H. Rezaie, and M. Ziaratban, "A morphological-based license plate location," in *Proceedings of the IEEE International Conference on Image Processing*, pp. I57–I60, IEEE, San Antonio, Tex, USA, September 2007.
- [8] Y.-P. Huang, C.-H. Chen, Y.-T. Chang, and F. E. Sandnes, "An intelligent strategy for checking the annual inspection status of motorcycles based on license plate recognition," *Expert Systems with Applications*, vol. 36, no. 5, pp. 9260–9267, 2009.
- [9] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. von Deneen, and P. Shi, "An algorithm for license plate recognition applied to intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 830–845, 2011.
- [10] K. Haneda and H. Hanaizumi, "A flexible method for recognizing four-digit numbers on a license-plate in a video scene," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT '12)*, pp. 112–116, IEEE, Athens, Ga, USA, March 2012.
- [11] W. Zhou, H. Li, Y. Lu, and Q. Tian, "Principal visual word discovery for automatic license plate detection," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 4269–4279, 2012.
- [12] A. M. Al-Ghaili, S. Mashohor, A. R. Ramli, and A. Ismail, "Vertical-edge-based car-license-plate detection method," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 26–38, 2013.

- [13] G.-S. Hsu, J.-C. Chen, and Y.-Z. Chung, "Application-oriented license plate recognition," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 2, pp. 552–561, 2013.
- [14] G. Abo Samra and F. Khalefah, "Localization of license plate number using dynamic image processing techniques and genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 244–257, 2014.

Research Article

A Multikernel-Like Learning Algorithm Based on Data Probability Distribution

Guo Niu,¹ Zhengming Ma,¹ and Shuyu Liu²

¹School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China

²Public Experimental Teaching Center, Sun Yat-sen University, Guangzhou 510006, China

Correspondence should be addressed to Shuyu Liu; lsy@mail.sysu.edu.cn

Received 22 February 2016; Accepted 9 May 2016

Academic Editor: Marco Perez-Cisneros

Copyright © 2016 Guo Niu 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 the machine learning based on kernel tricks, people often put one variable of a kernel function on the given samples to produce the basic functions of a solution space of learning problem. If the collection of the given samples deviates from the data distribution, the solution space spanned by these basic functions will also deviate from the real solution space of learning problem. In this paper a multikernel-like learning algorithm based on data probability distribution (MKDPD) is proposed, in which the parameters of a kernel function are locally adjusted according to the data probability distribution, and thus produces different kernel functions. These different kernel functions will generate different Reproducing Kernel Hilbert Spaces (RKHS). The direct sum of the subspaces of these RKHS constitutes the solution space of learning problem. Furthermore, based on the proposed MKDPD algorithm, a new algorithm for labeling new coming data is proposed, in which the basic functions are retrained according to the new coming data, while the coefficients of the retrained basic functions remained unchanged to label the new coming data. The experimental results presented in this paper show the effectiveness of the proposed algorithms.

1. Introduction

(a) *Data Spaces and Data Distributions.* Let X represent the data and Ω the data space. In mathematics, the data X can be regarded as a random variable/vector/matrix defined on the data space Ω . There will be different kinds of data on a data space. For example, the data space Ω can be the one consisting of all images of 512×512 pixels, while the data X_f represents all face images of 512×512 pixels and the data X_l represents all landscape images of 512×512 pixels; both of them are defined on the data space Ω but subject to different probability distributions.

If the data is regarded as a random variable, the samples of data can be then regarded as the concrete realization of the random variable. In machine learning, the samples of data can be exploited to estimate the probability distribution of data (the probabilistic modeling of data). There are a lots of researches on the probabilistic modeling of data [1–3].

(b) *Classification/Labels and Classifiers/Label Functions.* The classification of the data may be different when the data are

used in different applications. For example, let the data be the face images. In the application of identification recognition, the face images of the same person are all grouped into the same class, even though the expressions and postures of these face images may be different. However, in the application of expression recognition, the face images of the same expression are all grouped into the same class, even though these face images belong to different persons.

The classifier of data is a machine indicating the class to which a data point belongs. The classifiers of data are trained with the data samples [4, 5]. The classes of data are also called the labels of data and the classifiers of data are also called the label functions of data. In this paper, we adopt the terminology of labels and label functions of data.

(c) *Kernel Tricks in Machine Learning.* The applications of kernel tricks in machine learning can be roughly divided into two categories: the transformation of data spaces and the construction of label functions. In the transformation of data spaces, the kernel functions are used to transform data spaces into other spaces where the data can be linearly separated.

The famous Kernel PCA [6] and kernel Fisher discriminant (KFD) [7] belong to this category. In the construction of label functions, the kernel functions are used to serve as the basic functions of label functions. The famous manifold regularization learning [8, 9] belongs to this category. In this paper we address the problems involved in the construction of label functions.

In the construction of label functions, the label functions are expressed as $f(x) = \sum_{i=1}^{b+u} \alpha_i k(x, x_i)$, where $k(x, v)$ represents a kernel function, $\{x_1 \cdots x_b\}$ the labeled samples, and $\{x_{b+1} \cdots x_{b+u}\}$ the unlabeled samples. The coefficients $\vec{\alpha} = [\alpha_1 \cdots \alpha_{b+u}]$ can be derived from solving the following learning problem:

$$\begin{aligned} \vec{\alpha}^* &= \arg \min_{f \in H_X} \sum_{i=1}^b V(y_i, f(x_i)) + \kappa \|f\|_{H_X}^2 \\ &= \arg \min_{\vec{\alpha} \in R^{b+u}} \sum_{i=1}^b V\left(y_i, \sum_{j=1}^{b+u} \alpha_j k(x_i, x_j)\right) + \kappa \vec{\alpha}^T K_X \vec{\alpha}. \end{aligned} \quad (1)$$

In the above equation, $H_X = \text{span}\{k(x, x_i) \mid i = 1, \dots, b+u\}$ represents the solution space of the learning problem; $\{y_1 \cdots y_b\}$ represents the labels of the labeled samples $\{x_1 \cdots x_b\}$;

$$K_X = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_{b+u}) \\ \vdots & \ddots & \vdots \\ k(x_{b+u}, x_1) & \cdots & k(x_{b+u}, x_{b+u}) \end{bmatrix} \quad (2)$$

represents the kernel matrix; and $V(y_i, f(x_i))$ represents the cost function. We want to find a label function $f(x)$ which will make the cost as small as possible.

There are two kinds of properties about the data. The first kind of properties is the natural properties of data. The probability distributions of data are the examples of natural properties of data. The second kind of properties is the semantic properties of data. The data labels are the examples of semantic properties of data. It is clear that, in the framework of learning problem shown in (1), the semantic properties hidden in the data labels have been fully utilized, while the natural properties hidden in data probability distributions seem not to be deeply exploited. At present, the usual way to learn more information other than the data labels is to add various regularization terms to the cost function. For example, in the manifold regularization learning [8–11], a manifold regularization term is added to the cost function:

$$f^* = \arg \min_{f \in H_X} \sum_{i=1}^b V(y_i, f(x_i)) + \kappa \|f\|_{H_X}^2 + \lambda \|f\|_M^2, \quad (3)$$

where $\|f\|_M^2 = \vec{f}_X^T L_X \vec{f}_X$ is the so-called manifold regularization term, in which $\vec{f}_X = [f(x_1) \cdots f(x_{b+u})]$ and L_X is the Laplacian matrix reflecting the adjacency relations of data samples [12].

However, the addition of too many regularization terms to the cost function will make the learning problem complicated and difficult to solve. In this paper, rather than

adding regularization terms, an alternative way to exploit the data probability distribution in the learning problem is proposed. For the convenience of description, let us denote the kernel function as $k(x, z \mid \theta)$, where θ represents the parameter of the kernel function. In the proposed algorithm, the parameters of the basic functions $k(x, x_i \mid \theta_i)$ are adjusted based on the data distribution sample-by-sample; that is, $\theta_i = \theta(p(x_i))$, where $p(x)$ is the probability distribution of data, $i = 1, \dots, b+u$. These basic functions are then used to span the solution space of the learning problem: $H_D = \text{span}\{k(x, x_i \mid \theta_i) \mid i = 1, \dots, b+u\}$. It is clear that the probability distribution of data is integrated into the basic functions.

According to the theory of kernel functions, if $\theta_i \neq \theta_j$, then $k(x, z \mid \theta_i)$ and $k(x, z \mid \theta_j)$ are two different kernel functions and will generate two different RKHS. Now let H_i denote the RKHS generated by the kernel function $k(x, z \mid \theta_i)$, $i = 1, \dots, b+u$; then, theoretically speaking, the solution space H_D can be regarded as a subspace of the direct sum space $H_1 \oplus \cdots \oplus H_{b+u}$. Therefore, the proposed algorithm can be regarded as a kind of multikernel learning algorithm, but quite different from the commonly used multikernel learning algorithm [13–16]. Therefore, we call the proposed algorithm the multikernel-like learning algorithm based on data probability distribution, referred to as MKDPD algorithm.

How to label the new coming data (the out-of-samples) x is the key topic in machine learning [12, 17, 18]. There are two extreme algorithms. One algorithm uses the original label function f_{old} to label the new coming data x ; that is, the labels of the new coming data x are given by $f_{\text{old}}(x)$. This algorithm is best in efficiency, but worst in accuracy. Another algorithm regards the new coming data x as the unlabeled samples and mixes the new coming data x with the original samples and retrain a new label function f_{new} . The labels of the new coming data x are then given by $f_{\text{new}}(x)$. This latter algorithm is best in accuracy, but worst in efficiency. Various algorithms for labeling new coming data are the trade-off between these two extreme algorithms. In the proposed MKDPD algorithm, there are two learning processes. In the first learning process, the basic functions of label function are trained. In the second learning process, the weights of basic functions in the label function are solved. Accordingly, a new algorithm for labeling the new coming data is proposed. In the proposed algorithm, the new coming data will be exploited in the first learning process to retrain the basic functions, but the weights of basic functions remained unchanged and combined with the retrained basic functions to label new coming data. The proposed labeling algorithm achieves a better trade-off between the computational efficiency and accuracy.

The rest of the paper is arranged as follows: in Section 2, the literatures related to our work are reviewed briefly. In Section 3, the main theories of kernel functions and RKHS are introduced. In Section 4, the MKDPD algorithm is proposed. In Section 5, an MKDPD-based algorithm for labeling new coming data is proposed. In Section 6, the experimental results on toy and real-world data are presented to show the performance of the MKDPD algorithms. In Section 7, some conclusions are presented for reference.

2. Related Works

Learning from the given data is the main process to machine learning. So how to fully make use of the given samples is the key to the successful learning. In general, supervised learning has sufficient labeled samples and these kinds of algorithms are suitable for classification problems, such as the representative linear discriminant analysis (LDA) [19] and KFD [7]. In practice, a large number of samples are unlabeled, only a small number of samples are labeled. In this case, supervised learning algorithms do not effectively make use of the information hidden in unlabeled samples. To tackle the issue, semisupervised learning [18] is proposed and a wide range of semisupervised learning algorithms have been proposed and widely applied in many areas of machine learning.

In recent years, the study of semisupervised learning is not limited to the simple introduction of unlabeled data. Many researchers pay attention to exploit the intrinsic geometry of data and introduce the kernel learning to semisupervised methods. For example, manifold regularization learning proposed by Belkin et al. [8] exploits the underlying data structure by adding a manifold regular term to a general-purpose learner. And following it, a series of algorithms are proposed. Sindhvani et al. [20] proposed a linear MR (LMR) algorithm, in which a global linear mapping between the samples and their labels is constructed for labeling novel samples. Inspired by Gaussian fields and harmonic functions (GFHF) [21], local and global consistency (LGC) [22], and LMR [20], Nie et al. [23] extended LMR algorithm to flexible manifold embedding algorithm (FMA). FMA relaxes the hard linear constraint in LMR by adding a flexible regression residue. Geng et al. [10] proposed the ensemble manifold regularization (EMR) to deal with the aforementioned problems by learning an optimal graph Laplacian based on a set of given candidate graph Laplacian. With the sparse assumption, Fan et al. [24] replaced the manifold regularizer with a sparse regularizer under the MR framework. Luo et al. [11] applied MR framework to solve the problem of multilabel image classification by learning a discriminative subspace.

Introducing kernel trick to semisupervised learning methods is an important progress in machine learning. The kernel function [17] is either used to map input sample into a high dimensional kernel space for learning problem nonlinearization, or used to span a RKHS for the learning of label function. Take the MR learning as example, the label function (classifier function) in MR is a linear combination of single kernel function on labeled and unlabeled samples and the performance of MR algorithms strongly depends on this label function.

The theory of RKHS plays an important role in kernel methods and RKHS has found a wide range of applications such as minimum variance unbiased estimation of regression coefficients, least squares estimation of random variables, detection of signals in Gaussian noise, problems in optimal approximation [25]. Some RKHS-based learning algorithms appearing recently find applications to online learning with the problem of classification or regression [26–28], while others find applications to the classification of hyperspectral

images [29, 30]. Gurram and Kwon [29] achieved the weights for SVM separating hyperplanes by combining both local spectral and spatial information. Gu et al. [30] introduced the conception of Multiple-Kernel Hilbert Space (MKHS) to analyze spectral unmixing problems, and the resultant algorithm performs well in solving nonlinear problems.

In theory, RKHS can be generated with some specific functions called kernel functions such as Gaussian, Laplacian, and polynomial [31]. Modifying kernel functions is a way of improving the performance of kernel methods; for example, Wu and Amari [32] extended conformal transformation of kernel functions to improve the performance of Support Vector Machine classifiers, Gurram and Kwon [33] defined a new inner product to warp the RKHS structure to reflect the intrinsic geometry of the given samples, and the literatures [33, 34] obtained the best kernel parameters by calculating the derivatives of objective functions. The application of multiple kernels is hot topic in kernel methods and multikernel learning (MKL) [35] is a successful method, which enhances the interpretability of the classifier with a combination of base kernels and improves the performances of kernel methods. MKL algorithms have been widely investigated [13–16, 35–37] and the reviews of MKL algorithms can be found in [13, 14]. MKL offers a feasible scheme to ensemble multiple kernels, but high computational cost raised by optimization procedure is a bad limitation when it is used to process large-scale data and a number of kernels. Therefore, one main research direction in MKL is how to effectively solve the MKL problem. Lots of MKL algorithms have been proposed; for example, SimpleMKL [36] proposed by Rakotomamonjy et al. is one of the state-of-the-art algorithms used to solve MKL problem which is addressed by a simple subgradient descent method. However, the MKL task is still challenging because it must on one hand learn an optimal combination of multiple kernels and determine the optimal classifier in each iteration and on the other hand make sure that the two optimization procedures are feasible.

3. RKHS and Its Application to Machine Learning

3.1. RKHS. In machine learning RKHS are often used as the solution spaces of learning problems. The definition of RKHS is as follows.

Definition 1. Let $H = (S(\Omega), \langle \cdot, \cdot \rangle)$ be a Hilbert space; if there is a function $k : \Omega \times \Omega \rightarrow R$, such that

- (a) $\forall v \in \Omega, k(x, v) = k_v(x) \in S(\Omega)$;
- (b) $\forall f \in S(\Omega), f(x) = \langle f, k_x \rangle = \langle f(\cdot), k(\cdot, x) \rangle$,

then, H is said to be a Reproducing Kernel Hilbert Space (RKHS) and the function $k(x, z)$ is called the reproducing kernel of H .

Note that in $H = (S(\Omega), \langle \cdot, \cdot \rangle)$, Ω is a data space, $S(\Omega)$ is a linear space of functions defined on the data space Ω , and $\langle \cdot, \cdot \rangle$ is an inner product defined on $S(\Omega)$. According to the theory of RKHS, RKHS can be generated from a kernel function. The kernel function is defined as follows.

Definition 2. Let $k : \Omega \times \Omega \rightarrow R$ be a function such that

- (a) symmetric: $\forall x, v \in \Omega, k(x, v) = k(v, x)$;
- (b) positive definite: for all the positive integer n and all data $v_1, \dots, v_n \in \Omega$, the following matrix $K \in R^{n \times n}$ is positive definite:

$$\begin{bmatrix} k(v_1, v_1) & \cdots & k(v_1, v_n) \\ \vdots & \ddots & \vdots \\ k(v_n, v_1) & \cdots & k(v_n, v_n) \end{bmatrix}; \quad (4)$$

then the function $k(x, v)$ is called a kernel function.

A kernel function $k(x, v)$ can be used to generate a RKHS such that the kernel function is the reproducing kernel of RKHS. The generating procedure is as follows.

First, a linear space can be generated from the kernel function $k(x, v)$,

$$\begin{aligned} S_k(\Omega) &= \text{span} \{k_v(x) \mid k_v(x) = k(x, v), v \in \Omega\} \\ &= \left\{ f(x) \mid f(x) = \sum_{i=1}^n \alpha_i k(x, v_i), \alpha_i \in R, v_i \in \Omega, i \right. \\ &\quad \left. = 1, \dots, n, n \in N \right\}, \end{aligned} \quad (5)$$

where N is the set of all positive integers.

Second, an inner product $\langle \cdot, \cdot \rangle_k$ is defined on $S_k(\Omega)$; that is, for all $f, g \in S_k(\Omega)$, since $f(x) = \sum_{i=1}^n \alpha_i k(x, v_i)$ and $g(x) = \sum_{j=1}^n \beta_j k(x, u_j)$, the inner product of f and g is then defined as

$$\langle f, g \rangle_k = \sum_{i=1}^n \sum_{j=1}^n \alpha_i \beta_j k(v_i, u_j). \quad (6)$$

It is easy to prove that the definition of $\langle \cdot, \cdot \rangle_k$ meets the requirements of the inner product and, therefore, $H_k = (S_k(\Omega), \langle \cdot, \cdot \rangle_k)$ is an inner product space.

It is worth noting that for all $f \in S_k(\Omega)$, since $f(x) = \sum_{i=1}^n \alpha_i k(x, v_i)$

$$\begin{aligned} \langle f(\cdot), k(\cdot, x) \rangle_k &= \left\langle \sum_{i=1}^n \alpha_i k(\cdot, v_i), k(\cdot, x) \right\rangle_k \\ &= \sum_{i=1}^n \alpha_i \langle k(\cdot, v_i), k(\cdot, x) \rangle_k \\ &= \sum_{i=1}^n \alpha_i k(x, v_i) = f(x). \end{aligned} \quad (7)$$

That is to say, the functions in the inner space H_k can be reproduced with the kernel function $k(x, v)$.

Third, the inner space H_k can be completed if it is not completed. The completion of H_k , denoted by \overline{H}_k , is then a RKHS and the kernel function $k(x, v)$ is the reproducing kernel of \overline{H}_k . By the way, it can be seen from the completion that the inner space H_k is dense in the RKHS \overline{H}_k .

3.2. Solution Spaces of Machine Learning Problems. In practice, it is impossible to take the space H_k as the solution space of learning problem because it is infinite-dimensional. It is reasonable to require that the solution space be both finite-dimensional and sample-dependent. Thus, for the given samples $\{x_1, \dots, x_b, x_{b+1}, \dots, x_{b+u}\}$, a linear space can be generated as follows:

$$\begin{aligned} S_X(\Omega) &= \text{span} \{k(x, x_i) \mid i = 1, \dots, b+u\} \\ &= \left\{ \sum_{i=1}^{b+u} \alpha_i k(x, x_i) \mid \alpha_i \in R, i = 1, \dots, b+u \right\}. \end{aligned} \quad (8)$$

It is clear that $S_X(\Omega)$ is both finite-dimensional and sample-dependent. Furthermore, $S_X(\Omega)$ is exactly a subspace of $S_k(\Omega)$ and therefore $H_X = (S_X(\Omega), \langle \cdot, \cdot \rangle_k)$ is a subspace of H_k . Since H_X is finite-dimensional, then it is complete; that is, H_X is a Hilbert space. However, H_X is no longer a RKHS.

For all functions $f, g \in S_X(\Omega)$, since $f(x) = \sum_{i=1}^{b+u} \alpha_i k(x, x_i)$ and $g(x) = \sum_{j=1}^{b+u} \beta_j k(x, x_j)$, then, according to (6), we have

$$\langle f, g \rangle_k = \sum_{i=1}^{b+u} \sum_{j=1}^{b+u} \alpha_i \beta_j k(x_i, x_j) = \vec{\alpha}^T K_X \vec{\beta}, \quad (9)$$

where $\vec{\alpha} = [\alpha_1 \cdots \alpha_{b+u}]^T$, $\vec{\beta} = [\beta_1 \cdots \beta_{b+u}]^T$, and

$$K_X = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_{b+u}) \\ \vdots & \ddots & \vdots \\ k(x_{b+u}, x_1) & \cdots & k(x_{b+u}, x_{b+u}) \end{bmatrix}. \quad (10)$$

Since the matrix K_X is symmetric and positive definite, the inner product on $S_X(\Omega)$ can be defined by itself, not necessarily inherited from $\langle \cdot, \cdot \rangle_k$. In fact, for all $f, g \in S_X(\Omega)$, the inner product can be defined as

$$\langle f, g \rangle_X = \vec{\alpha}^T K_X \vec{\beta}. \quad (11)$$

It can be easily proven that the definition of $\langle \cdot, \cdot \rangle_X$ meets the requirements of inner product and therefore $H_X = (S_X(\Omega), \langle \cdot, \cdot \rangle_X)$ is an inner space. Again, since $S_X(\Omega)$ is finite-dimensional, H_X is complete. In machine learning, it is the space H_X that is taken as the solution space of learning problem.

4. A Multikernel-Like Learning Algorithm Based on Data Probability Distribution MKDPD

4.1. Motivation. As shown in (5), the space of label functions is as follows:

$$S_X(\Omega) = \text{span} \{k(x, x_i) \mid i = 1, \dots, b+u\}. \quad (12)$$

This means that the functions $\{k(x, x_i) \mid i = 1, \dots, b+u\}$ play the role of basic functions of $S_X(\Omega)$. Obviously, these basic functions are only dependent on the locations of given

samples and seem too simple to adapt to various probability distributions of data. Take Gaussian kernel function $k(u, v) = \exp^{-\|u-v\|^2/2\sigma^2}$ as an example, the basic functions $\{k(x, x_i) \mid i = 1, \dots, b+u\}$ generated from Gaussian kernel function are identical with each other, only different in the locations of data space. A basic function can be derived from another basic function only by translation in the data space Ω . In fact, if $i \neq j$, then

$$\begin{aligned} k_i(x - x_j + x_i) &= \exp^{-\|x - x_j + x_i - x_i\|^2/2\sigma^2} = k(x, x_j) \\ &= k_j(x). \end{aligned} \quad (13)$$

Furthermore, since $f(x) = \sum_{i=1}^{b+u} \alpha_i k(x, x_i)$, then for all $x \in \Omega$ with $p(x) \neq 0$, $f(x)$ should give the label of x . This means that $\text{sup } p(p) \subseteq \cup_{i=1}^{b+u} \text{sup } p(k(\cdot, x_i))$, where $\text{sup } p(p)$ is the support of $p(x)$ and $\text{sup } p(k(\cdot, x_i))$ is the support of $k(x, x_i)$; that is,

$$\begin{aligned} \text{sup } p(p) &= \{x \mid x \in \Omega, p(x) \neq 0\}, \\ \text{sup } p(k(\cdot, x_i)) &= \{x \mid x \in \Omega, k(x, x_i) \neq 0\}. \end{aligned} \quad (14)$$

If the relation $\text{sup } p(p) \subseteq \cup_{i=1}^{b+u} \text{sup } p(k(\cdot, x_i))$ is not true, there would be $x \in \Omega$ such that $p(x) \neq 0$, but $f(x) = \sum_{i=1}^{b+u} \alpha_i k(x, x_i) = 0$; that is, $f(x)$ cannot give the label of x .

However, the union $\cup_{i=1}^{b+u} \text{sup } p(k(\cdot, x_i))$ is dependent on the locations of the given data samples $\{x_1 \dots x_{b+u}\}$, not dependent on the data probability distribution $p(x)$. In practice, kernel functions are often compactly supported and the data are not evenly distributed over the data space. In these cases, label function $f(x)$ will be overfitted in the areas where too many data samples are collected, or underfitted in the areas where there are too few data samples collected, or not fitted at all in the area where the union $\cup_{i=1}^{b+u} \text{sup } p(k(\cdot, x_i))$ fails to cover.

Based on the above considerations, a learning algorithm based on the data probability distribution is proposed in this paper. In the proposed algorithm, the union is not only dependent on the locations of the given samples, but also dependent on the data probability distribution.

4.2. Construction of Solution Spaces. For the convenience of description, let $k(x, v \mid \theta)$ denote the kernel function, where θ represents the parameter of the kernel function. Thus, for the given data samples $\{x_1 \dots x_{b+u}\}$ and data probability distribution $p(x)$, the basic functions of solution space generated from the kernel function $k(x, v \mid \theta)$ are expressed as $k(x, x_i \mid \theta_i)$, where $\theta_i = \theta_{p(i)}$, $i = 1 \dots b+u$.

With these basic functions, we can span a linear space $S_D(\Omega)$ as follows:

$$\begin{aligned} S_D(\Omega) &= \text{span} \{k(x, x_i \mid \theta_i) \mid i = 1, \dots, b+u\} \\ &= \left\{ \sum_{i=1}^{b+u} \alpha_i k(x, x_i \mid \theta_i) \mid i = 1, \dots, b+u \right\}. \end{aligned} \quad (15)$$

It is clear that $S_D(\Omega)$ is a finite-dimensional linear space. Further, in order to define an inner product on $S_D(\Omega)$, we need to define a symmetric and positive definite matrix first:

$$M_D = K_D^T K_D + \rho I, \quad (16)$$

where $\rho, I \in R^{(b+u) \times (b+u)}$ is a unit matrix and

$$K_D = \begin{bmatrix} k(x_1, x_1 \mid \theta_1) & \dots & k(x_1, x_{b+u} \mid \theta_{b+u}) \\ \vdots & \ddots & \vdots \\ k(x_{b+u}, x_1 \mid \theta_1) & \dots & k(x_{b+u}, x_{b+u} \mid \theta_{b+u}) \end{bmatrix}. \quad (17)$$

Note that, since $k(x_i, x_j \mid \theta_j) \neq k(x_j, x_i \mid \theta_i)$, $i \neq j$, K_D is not symmetric and positive definite. However, M_D is symmetric and definite positive and can be used to define an inner product $\langle \cdot, \cdot \rangle_D$ on $S_D(\Omega)$: for all $f, g \in S_D(\Omega)$, since $f(x) = \sum_{i=1}^{b+u} \alpha_i k(x, x_i \mid \theta_i)$ and $g(x) = \sum_{j=1}^{b+u} \beta_j k(x, x_j \mid \theta_j)$, then

$$\langle f, g \rangle_D = \vec{\alpha}^T M_D \vec{\beta}, \quad (18)$$

where $\vec{\alpha} = [\alpha_1 \dots \alpha_{b+u}]^T$, $\vec{\beta} = [\beta_1 \dots \beta_{b+u}]^T$.

It can be easily proven that $\langle \cdot, \cdot \rangle_D$ meets the requirements of inner product and therefore $H_D = (S_D(\Omega), \langle \cdot, \cdot \rangle_D)$ is an inner product space. Furthermore, since $S_D(\Omega)$ is finite-dimensional, H_D is then complete; that is, H_D is a Hilbert space. However, it is worth noting that H_D is neither a RKHS, nor a subspace of H_k . Recall that although H_X is not RKHS, H_X is a subspace of H_k .

In the proposed algorithm, H_D is taken as the solution space of learning problem:

$$f^* = \arg \min_{f \in H_D} \sum_{i=1}^b V(y_i, f(x_i)) + \kappa \|f\|_{H_D}^2 + \lambda \|f\|_M^2. \quad (19)$$

Below we explain the rationality of the definition $\langle \cdot, \cdot \rangle_D$:

- (1) If $\langle \cdot, \cdot \rangle_D$ is an inner product of $S_D(\Omega)$, according to the linearity of inner product, for all $f, g \in S_D(\Omega)$, we have

$$\begin{aligned} \langle f, g \rangle &= \left\langle \sum_{i=1}^{b+u} \alpha_i k(\cdot, x_i \mid \theta_i), \sum_{j=1}^{b+u} \beta_j k(\cdot, x_j \mid \theta_j) \right\rangle \\ &= \sum_{i=1}^{b+u} \sum_{j=1}^{b+u} \alpha_i \beta_j \langle k(\cdot, x_i \mid \theta_i), k(\cdot, x_j \mid \theta_j) \rangle. \end{aligned} \quad (20)$$

Usually, the inner product of functional space is often defined as the integral of product of functions; therefore we have

$$\begin{aligned} &\langle k(\cdot, x_i \mid \theta_i), k(\cdot, x_j \mid \theta_j) \rangle \\ &= \int_{\Omega} k(x, x_i \mid \theta_i) k(x, x_j \mid \theta_j) dx \\ &\approx \sum_{h=1}^{b+u} k(x_h, x_i \mid \theta_i) k(x_h, x_j \mid \theta_j). \end{aligned} \quad (21)$$

Substituting (21) into (20) gives

$$\begin{aligned} \langle f, g \rangle &\approx \sum_{i=1}^{b+u} \sum_{j=1}^{b+u} \sum_{h=1}^{b+u} \alpha_i \beta_j k(x_h, x_i | \theta_i) k_j(x_h, x_j | \theta_j) \\ &= \tilde{\alpha}^T K_D^T K_D \tilde{\beta}. \end{aligned} \quad (22)$$

However the matrix $K_D^T K_D$ is only positive semidefinite and cannot be used to define an inner product. This problem can be easily solved by adding a regularization term ρI , where $I \in R^{(b+u) \times (b+u)}$ is the unit matrix and ρ is the regularization parameter: $M_D = K_D^T K_D + \rho I$.

The matrix M_D is now symmetric and positive definite and can be used to define an inner product on $S_D(\Omega)$:

$$\begin{aligned} \langle f, g \rangle_D &= \tilde{\alpha}^T M_D \tilde{\beta} = \tilde{\alpha}^T K_D^T K_D \tilde{\beta} + \rho \tilde{\alpha}^T \tilde{\beta} \\ &\approx \tilde{\alpha}^T K_D^T K_D \tilde{\beta} = \langle f, g \rangle. \end{aligned} \quad (23)$$

The regularization parameter ρ can also alleviate the ill-condition of the matrix $K_D^T K_D$ and reduce the error stemming from the substitution of integral with summation in (29).

- (2) If the data probability distribution $p(x)$ is uniform, that is, $p(x)$ is constant on the support $\text{sup } p(p)$, then $\theta_i = \theta(p(x_i))$, $i = 1, \dots, b+u$. In this case, we have

$$\langle k(\cdot, x_i | \theta_i), k(\cdot, x_j | \theta_j) \rangle = k(x_i, x_j | \theta), \quad (24)$$

$$\langle f, g \rangle = \langle f, g \rangle_X. \quad (25)$$

Combining (23) and (25) will give the following result:

$$\langle f, g \rangle_D \approx \langle f, g \rangle = \langle f, g \rangle_X. \quad (26)$$

This means that if the parameter θ of the kernel function $k(x, x_i | \theta)$ does not adjust sample by sample, then $H_D = H_X$.

4.3. Analytic Solutions to Learning Problems

4.3.1. Analytic Solutions to Two-Class Learning Problems. In the proposed algorithm, the Hilbert space H_D is taken as the solution space of learning problems. Then for all $f \in H_D$, $f(x) = \sum_{i=1}^{b+u} \alpha_i k(x, x_i | \theta_i)$, we have

$$\tilde{f}_D = [f(x_1) \cdots f(x_{b+u})]^T = K_D \tilde{\alpha}, \quad (27)$$

$\|f\|_D^2 = \tilde{\alpha}^T M \tilde{\alpha}$, $\|f\|_M^2 = \tilde{\alpha}^T K_D^T L_X K_D \tilde{\alpha}$. Based on the above results, the problem shown in (19) can be simplified as follows:

$$\begin{aligned} \tilde{\alpha}^* &= \arg \min_{\tilde{\alpha} \in R^{b+u}} \left\{ \sum_{i=1}^b V \left(y_i, \sum_{i=1}^{b+u} \alpha_i k(x_i, x_j | \theta_j) \right) \right. \\ &\quad \left. + \tilde{\alpha}^T (\kappa M_D + \lambda K_D^T L_X K_D) \tilde{\alpha} \right\}. \end{aligned} \quad (28)$$

Furthermore, if the cost function $V(y, f(x))$ is set to be the square error function, that is, $V(y, f(x)) = (y - f(x))^2$, we have

$$\begin{aligned} \tilde{\alpha}^* &= \arg \min_{\tilde{\alpha} \in R^{b+u}} \tilde{y}^T \tilde{y} - 2 \tilde{y}^T S K_D \tilde{\alpha} \\ &\quad + \tilde{\alpha}^T (K_D^T S^T S K_D + \kappa M_D + \lambda K_D^T L_X K_D) \tilde{\alpha} \\ &= \arg \min_{\tilde{\alpha} \in R^{b+u}} \tilde{y}^T \tilde{y} - 2 \tilde{b}_D^T \tilde{\alpha} + \tilde{\alpha}^T A_D \tilde{\alpha} = A_D^{-1} \tilde{b}_D, \end{aligned} \quad (29)$$

where S is the selection matrix, $\tilde{b}_D = K_D^T S^T \tilde{y}$, and $A_D = K_D^T S^T S K_D + \kappa M_D + \lambda K_D^T L_X K_D$. Note that the matrix A_D is a symmetric and positive definite matrix.

4.3.2. Analytic Solutions to Multiclass Learning Problems. In principle, the deduction shown above is also suitable for the multiclass learning problems. In fact, for the data sample x_i , its label y_i can take different values to indicate the different classes to which the data sample x_i belongs. However, in practice, the different values of y_i may be too close to facilitate the optimization calculation. Therefore, for the multiclass problems, we adopt another way to indicate the data labels.

For the data sample x_i , let its label \tilde{y}_i be a C -dimensional vector, where C is the number of classes. If the data sample x_i belongs to the c th class, then the c th component of \tilde{y}_i is set to be 1 while the other components of \tilde{y}_i are set to zero, where $c = 1, \dots, C$. Furthermore, a label function $f_c(x)$ is set to describe the probability that the data x belongs to the c th class. Based on these notations, the multiclass problem can be expressed as follows:

$$\begin{aligned} f^* &= \arg \min_f \sum_{i=1}^b \|\tilde{y}_i - \tilde{g}_i\|^2 + \kappa \sum_{c=1}^C \|\tilde{f}_c\|_{H_D}^2 \\ &\quad + \lambda \sum_{c=1}^C \tilde{f}_c^T L_X \tilde{f}_c, \end{aligned} \quad (30)$$

where

$$\begin{aligned} f &= (f_1, \dots, f_C), \\ \tilde{g}_i &= \begin{bmatrix} f_1(x_i) \\ \vdots \\ f_C(x_i) \end{bmatrix}, \\ \tilde{f}_c &= \begin{bmatrix} f_c(x_1) \\ \vdots \\ f_c(x_i) \end{bmatrix}, \end{aligned} \quad (31)$$

$i = 1, \dots, l$.

We first calculate the term $\sum_{c=1}^C \|\tilde{f}_c\|_{H_D}^2$. Since $f_c(x) \in S_D(\Omega)$, then $f_c(x) = \sum_{i=1}^{b+u} \alpha_i^c k(x, x_i | \theta_i)$ and

$$\sum_{c=1}^C \|\tilde{f}_c\|_{H_D}^2 = \sum_{c=1}^C \tilde{\alpha}_c^T M_D \tilde{\alpha}_c = \text{Tr}(A^T M_D A), \quad (32)$$

where $\tilde{\alpha}_c = [\alpha_1^c, \dots, \alpha_{b+u}^c]^T$ and $A = [\tilde{\alpha}_1, \dots, \tilde{\alpha}_C]$.

Secondly, we calculate the term $\sum_{c=1}^C \vec{f}_c^T L_X \vec{f}_c$. Since $\vec{f}_c = K_D \vec{\alpha}_c$ then

$$\sum_{c=1}^C \vec{f}_c^T L_X \vec{f}_c = \text{trace} \left(A^T K_D^T L_X K_D A \right). \quad (33)$$

Thirdly, we calculate the term $\sum_{i=1}^b \|\vec{y}_i - \vec{g}_i\|^2$. Let $Y = [\vec{y}_1, \dots, \vec{y}_b]$, $G = [\vec{g}_1, \dots, \vec{g}_b]$, and $F = [\vec{f}_1, \dots, \vec{f}_C] = K_D A$; then

$$\begin{aligned} \sum_{i=1}^b \|\vec{y}_i - \vec{g}_i\|^2 &= \|Y - G\|^2 \\ &= \text{trace} \left(Y^T Y - 2YSK_D A + A^T K_D^T S^T S K_D A \right). \end{aligned} \quad (34)$$

Again, the matrix S is a selection matrix such that $G = F^T S^T$.

At last, substituting (32), (33), and (34) into (30) will give the following result:

$$\begin{aligned} A^* &= \arg \min_A \text{trace} \left(Y Y^T - 2YSK_D A \right. \\ &\quad \left. + A^T \left(K_D^T S^T K_D + \kappa M_D + \lambda K_D^T L K_D \right) A \right) \\ &= \left(K_D^T \left(S^T S + \lambda L + \kappa I \right) K_D + \rho I \right)^{-1} K_D^T S^T Y^T. \end{aligned} \quad (35)$$

4.4. The Framework of Multikernel-Like Learning Algorithms.

In the algorithm proposed in this paper, the label function $f(x)$ is set to be $f(x) = \sum_{i=1}^{b+u} \alpha_i k(x, x_i | \theta_i)$, where the parameter θ_i is adjusted according to the data probability distribution $p(x)$ on the data sample x_i . In general, the data probability distribution is not uniform and therefore the parameter θ_i will be different sample by sample. As a result, the functions $k(x, v | \theta_1), \dots, k(x, v | \theta_{b+u})$ are different kernel functions and will produce different RKHS. In this sense, the algorithm proposed in this paper can be regarded as a kind of multikernel learning algorithms, but quite different from the commonly used multikernel learning algorithm.

In the commonly used multikernel learning algorithms, the multikernel function is a linear combination of multiple basic kernel functions: $k_{\text{MK}}(x, v) = \sum_{j=1}^M \beta_j k_j(x, v)$, where the functions k_1, \dots, k_M are called basic kernel functions, while the function $k_{\text{MK}}(x, v)$ is called the multikernel function. Since the basic kernel functions are symmetric and positive definite, it can be easily proven that the multikernel function is also symmetric and positive definite. Therefore the label function $f_{\text{MK}}(x)$ based on the multikernel function can be expressed as

$$\begin{aligned} f_{\text{MK}}(x) &= \sum_{i=1}^{b+u} \alpha_i k_{\text{MK}}(x, x_i) = \sum_{i=1}^{b+u} \alpha_i \sum_{j=1}^M \beta_j k_j(x, x_i) \\ &= \sum_{i=1}^{b+u} \sum_{j=1}^M \alpha_i \beta_j k_j(x, x_i), \end{aligned} \quad (36)$$

where the coefficients $\alpha_1, \dots, \alpha_{b+u}$ and β_1, \dots, β_M are determined through machine learning.

If we follow the ideas of the commonly used multikernel learning algorithms and regard the functions $k(x, v | \theta_1), \dots, k(x, v | \theta_{b+u})$ as the basic kernel functions, then the multikernel function becomes $k_{\text{MK}}(x, v) = \sum_{j=1}^M \beta_j k(x, v | \theta_j)$. Thus, according to (36), the label function $f_{\text{MK}}(x)$ based on the multikernel function becomes

$$\begin{aligned} f_{\text{MK}}(x) &= \sum_{i=1}^{b+u} \alpha_i k_{\text{MK}}(x, x_i) = \sum_{i=1}^{b+u} \alpha_i \sum_{j=1}^M \beta_j k(x, x_i | \theta_j) \\ &\neq \sum_{i=1}^{b+u} \alpha_i k(x, x_i | \theta_i) = f(x). \end{aligned} \quad (37)$$

It can be seen from (37) that, no matter how to adjust the coefficients β_1, \dots, β_M , it is impossible to make $f_{\text{MK}}(x) \neq f(x)$. From the perspective of solution spaces, in the solution space of $f_{\text{MK}}(x)$, there are $b+u$ functions $k(x, x_i | \theta_1), \dots, k(x, x_i | \theta_{b+u})$ around the data sample x_i , while in the solution space of $f(x)$, there is only one function $k(x, x_i | \theta_i)$ around the data sample x_i . Therefore the algorithm proposed in this paper is quite different from the commonly used multikernel learning algorithm.

Nevertheless, the algorithm proposed in this paper still belongs to the realm of multikernel learning. As stated above, the functions $k(x, x_i | \theta_1), \dots, k(x, x_i | \theta_{b+u})$ are different kernel functions and can produce different RKHS: H_1, \dots, H_{b+u} . Now let $V_i = \{\alpha k(x, x_i | \theta_i) | \alpha \in \mathbb{R}\}$, as stated in Section 3.1; V_i is then a 1-dimensional subspace of H_i . Furthermore, the direct sum of these subspaces turns out to be

$$V = V_1 \oplus \dots \oplus V_{b+u} = \left\{ \sum_{i=1}^{b+u} \alpha_i (x, x_i | \theta_i) | \alpha_i \in \mathbb{R} \right\}. \quad (38)$$

Obviously the direct sum of these subspaces is the solution space H_D .

Due to the fact that our algorithm is different from the commonly used multikernel learning algorithm, but still involved in multiple kernel functions, our algorithm is called multikernel-like algorithm.

5. An MKDPD-Based Algorithm for Labeling New Coming Data

5.1. Problems. How to label new coming data $\{x_1^{\text{new}}, \dots, x_n^{\text{new}}\}$ has been a hot topic in machine learning, where $n \geq 1$ represents the number of new coming data. Generally speaking, there are two extreme methods for labeling new coming data: relearning methods and unlearning methods.

The relearning methods regard the new coming data as unlabeled data samples and mix them with the original data samples to form new data samples: $\{x_1, \dots, x_{b+u}, x_{b+u+1}, \dots, x_{b+u+n}\}$, where $x_{b+u+j} = x_j^{\text{new}}$, $j = 1, \dots, n$. Based on these new data samples, the methods relearn the new coefficients $\{\alpha_1^{\text{new}}, \dots, \alpha_{b+u+n}^{\text{new}}\}$ of the label function f . The label of the new coming data x_j^{new} is then given by $f_{\text{MR}}^{\text{re}}(x_j^{\text{new}}) = \sum_{i=1}^{b+u+n} \alpha_i^{\text{new}} k(x_j^{\text{new}}, x_i)$.

The unlearning methods make use of the original coefficients $\{\alpha_1^{\text{old}}, \dots, \alpha_{b+u}^{\text{old}}\}$ of label function f to label new coming data: $f_{\text{MR}}^{\text{un}}(x_j^{\text{new}}) = \sum_{i=1}^{b+u} \alpha_i^{\text{old}} k(x_j^{\text{new}}, x_i)$.

Obviously, in terms of accuracy, the relearning methods perform best, while the unlearning methods perform worst. In terms of efficiency, the unlearning methods perform best, while the relearning methods perform worst. For years the researchers have been hovering between these two extreme methods and try to find the trade-offs between accuracy and efficiency.

5.2. A MKDPD-Based Algorithm for Labeling New Coming Data. As stated in Section 4, there are two times of learning in the MKDPD algorithm. In the first time of learning, the MKDPD algorithm has to adjust the parameters of kernel functions according to data probability distribution. In the second time of learning, the MKDPD algorithm has to determine the coefficients of label functions. Therefore, in the framework of the MKDPD algorithm, there are at least three ways to label new coming data:

- (1) The MKDPD-based relearning method: $f_{\text{MKDPD}}^{\text{re}}(x_j^{\text{new}}) = \sum_{i=1}^{b+u+n} \alpha_i^{\text{new}} k(x_j^{\text{new}}, x_i | \theta_i^{\text{new}})$, $j = 1, \dots, n$. Obviously, the MKDPD-based relearning method can achieve the best accuracy but perform worst in efficiency because the MKDPD-based relearning method has to calculate both the parameters θ^{new} and coefficients α^{new} .
- (2) The MKDPD-based unlearning method: $f_{\text{MKDPD}}^{\text{un}}(x_j^{\text{new}}) = \sum_{i=1}^{b+u} \alpha_i^{\text{old}} k(x_j^{\text{new}}, x_i | \theta_i^{\text{old}})$. Obviously, the MKDPD-based unlearning method can achieve the best efficiency but perform worst in accuracy because the unlearning method does not calculate the parameters θ^{new} and coefficients α^{new} either.
- (3) The MKDPD-based semilearning method: $f_{\text{MKDPD}}^{\text{semi}}(x_j^{\text{new}}) = \sum_{i=1}^{b+u} \alpha_i^{\text{old}} k(x_j^{\text{new}}, x_i | \theta_i^{\text{new}})$. The MKDPD-based semilearning method regards the new coming data as unlabeled data samples and mixes them with the original data samples to retrain the new parameters θ^{new} of kernel functions. However, the coefficients α^{old} remained unchanged and combined with the retrained kernel functions to label the new coming data. The MKDPD-based semilearning method takes full advantage of two times of learning in the MKDPD algorithm and achieves a better trade-off between computational accuracy and efficiency.

5.3. Error and Efficiency Analysis

5.3.1. Experimental Data and Experimental Settings. We test our algorithm in the framework of manifold regularization and therefore the experimental data are downloaded from the website of manifold regularization (http://manifold.cs.uchicago.edu/manifold_regularization/manifold.html). There are a total of 400 sets of data collected from two half-moons, 200

TABLE 1: Average error rates (%) of four algorithms on testing samples with different number of tests.

| Algorithm | Train times | | | |
|----------------------------------|-------------|----------|----------|----------|
| | $N = 30$ | $N = 50$ | $N = 70$ | $N = 90$ |
| $f_{\text{MR}}^{\text{un}}$ | 0.8500 | 0.8800 | 0.7286 | 1.0167 |
| $f_{\text{MKDPD}}^{\text{un}}$ | 0.2333 | 0.2700 | 0.2786 | 0.2722 |
| $f_{\text{MKDPD}}^{\text{semi}}$ | 0.1667 | 0.2200 | 0.19297 | 0.2000 |
| $f_{\text{MKDPD}}^{\text{re}}$ | 0 | 0 | 0 | 0.0333 |

sets of data from one half-moon, and another 200 sets of data from another half-moon.

We randomly take 1 set of datum as labeled sample and 99 sets of data as unlabeled samples from each half-moon. The remaining 200 sets of data are taken as new coming data for labeling.

In order to alleviate the effect of random sampling on the objectivity of the experimental results, the random sampling has been done for N times and each random sampling will produce an experimental result. The average of N experimental results is taken as the end result. N is set to be 30, 50, 70, and 90, respectively (Table 1).

5.3.2. Error Analysis. Table 1 lists the error rates of various algorithms for labeling new coming data. Not surprisingly, the order of error rates is $f_{\text{MKDPD}}^{\text{re}} \leq f_{\text{MKDPD}}^{\text{semi}} \leq f_{\text{MKDPD}}^{\text{un}}$. This order coincides with the amount of information exploited by these algorithms from the new coming data.

In addition, the error rate of $f_{\text{MKDPD}}^{\text{un}}$ is smaller than that of $f_{\text{MR}}^{\text{un}}$, where $f_{\text{MKDPD}}^{\text{un}} = \sum_{i=1}^{b+u} \alpha_i^{\text{old}} k(x, x_i | \theta_i^{\text{old}})$ and $f_{\text{MR}}^{\text{un}} = \sum_{i=1}^{b+u} \alpha_i^{\text{old}} k(x, x_i)$. It can be seen from the formulae of $f_{\text{MKDPD}}^{\text{un}}$ and $f_{\text{MR}}^{\text{un}}$ that the basic functions $k(x, x_i | \theta_i^{\text{old}})$ of $f_{\text{MKDPD}}^{\text{un}}$ exploit not only the locations of samples, but also the probabilities of data on the samples, while the basic functions $k(x, x_i)$ of $f_{\text{MR}}^{\text{un}}$ exploit only the locations of samples.

Figure 1(a) shows the error rates of various algorithms change along with the number of new coming data. Again, the error rate of $f_{\text{MKDPD}}^{\text{semi}}$ is between those of $f_{\text{MKDPD}}^{\text{un}}$ and $f_{\text{MKDPD}}^{\text{re}}$.

5.3.3. Efficiency Analysis. Figure 1(b) shows the runtime of various algorithms in labeling the new coming data. It can be seen from Figure 1(b) that the runtime of $f_{\text{MKDPD}}^{\text{re}}$ increases exponentially along with the number of new coming data, while the runtime of $f_{\text{MKDPD}}^{\text{un}}$, $f_{\text{MKDPD}}^{\text{semi}}$, and $f_{\text{MR}}^{\text{un}}$ almost remains unchanged.

Since both $f_{\text{MKDPD}}^{\text{un}}$ and $f_{\text{MR}}^{\text{un}}$ make use of the original parameters θ^{old} and α^{old} to label the new coming data, the runtime of $f_{\text{MKDPD}}^{\text{un}}$ and $f_{\text{MR}}^{\text{un}}$ will not change no matter how many new coming data are coming. However, in the proposed algorithm $f_{\text{MKDPD}}^{\text{semi}}$, although the parameters θ are retrained from θ^{old} to θ^{new} according to the new coming data, the runtime of $f_{\text{MKDPD}}^{\text{semi}}$ almost remains unchanged along with the number of new coming data. The means that $f_{\text{MKDPD}}^{\text{semi}}$ achieves a certain amount of accuracy without increasing its runtime.

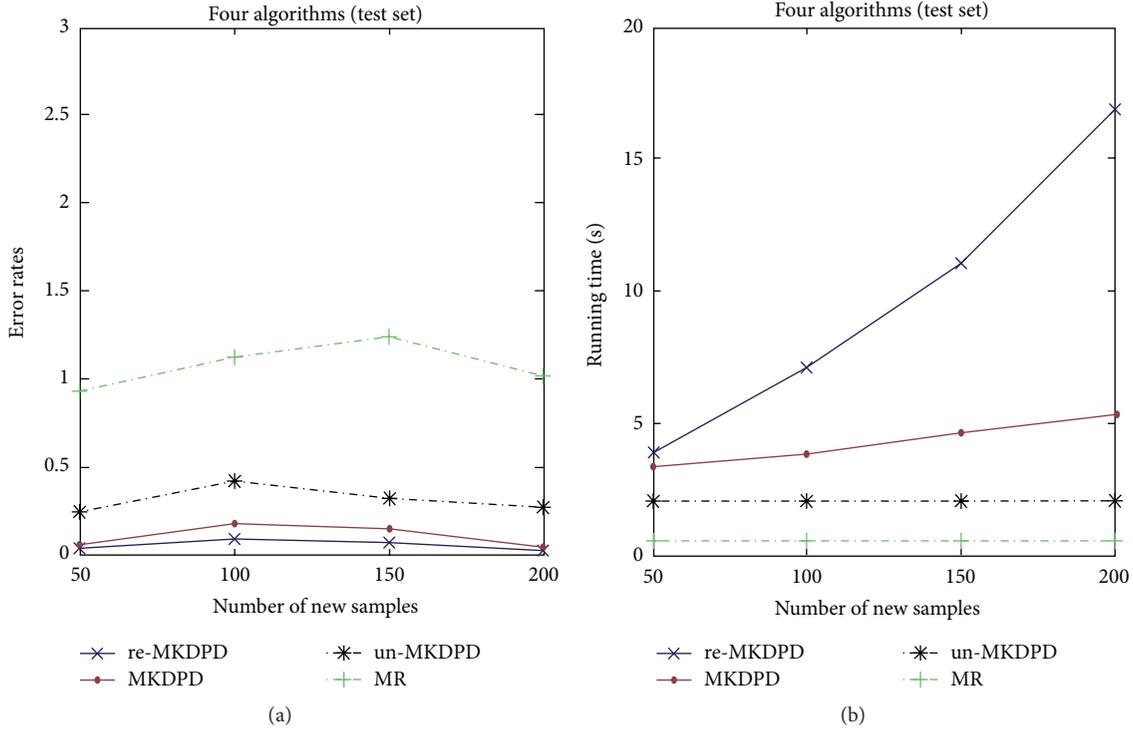


FIGURE 1: (a) shows the error rates of the four algorithms (f_{MR}^{un} , f_{MKDPD}^{un} , f_{MKDPD}^{semi} , and f_{MKDPD}^{re}) on the testing set, and the number of new coming samples is 50, 100, 150, and 200, respectively. (b) reports the computation time of the four algorithms with the increasing of the new coming sample (from 50 to 200).

6. Experiments

6.1. Adjustment of Parameters of Kernel Functions. In the proposed MKDPD algorithm, the basic functions of label function $f(x)$ are set to be $k(x, x_i | \theta_i)$, where $\theta_i = \theta(p(x_i))$, $i = 1, \dots, b + u$. The schemes of how to adjust the parameters θ_i are open. People can adopt various schemes according to their specific applications. No matter how to adjust the parameters θ_i , the structures of analytic solutions shown in (19) will not change in the framework of the proposed MKDPD algorithm. In the experiments presented in this paper, the scheme of adjusting the parameters is based on the following considerations;

- (1) The parameters θ_i should be adjusted so as to make $\sup p(p) \subseteq \cup_{i=1}^{b+u} \sup p(k(\cdot, x_i | \theta_i))$. In this way, for all $x \in \Omega$ with $p(x) \neq 0$, the label function $f(x)$ can give the label of x .
- (2) If the value of $p(x_i)$ is large, the number of samples gathering in the neighborhood of x_i will be large too because they are more likely to be collected. In order to prevent data overfitting in the area, it is reasonable to adjust the parameter θ_i to reduce the scope of the support $\sup p(k(\cdot, x_i | \theta_i))$. Conversely, if the value of $p(x_i)$ is small, the number of samples will be small too because they are more unlikely to be collected. In order to prevent data underfitting in the area, it is reasonable to adjust the parameter to expand the scope of the support $\sup p(k(\cdot, x_i | \theta_i))$. This means

that the probability $p(x_i)$ is inversely proportional to the scope of the support.

- (3) In the following experiments, we adopt Gaussian kernel functions $k(x, x_i | \theta_i) = \exp^{-\|x-x_i\|^2/2\theta_i^2}$. The Gaussian kernel function can be regarded as a compactly support function, the sample x_i is its center, and $3\theta_i$ is often regarded as its effective radius. Therefore, the parameter θ_i is proportional to the scope of the support $\sup p(k(\cdot, x_i | \theta_i))$, or inversely proportional to the probability $p(x_i)$; that is, $\theta_i = \eta_i / p(x_i)$, where η_i is an adjustable parameter. In our experiments, the probability $p(x_i)$ of sample x_i is set to $p(x_i) = p_i(x_i) / \sum_i p_i(x_i)$, where $p_i(x_i) = \varepsilon(x_i) / (b + u)$ and $\varepsilon(x_i)$ is the number of neighbors of x_i .

6.2. Experiments on Synthetic Dataset (Two Moons Dataset). The synthetic dataset is the Two Moons Dataset, which has already been used in the experiments in Section 5.

The Two Moons dataset contains 400 sets of data non-evenly collected from two half-moons, where 200 sets of data are collected from one half-moon and the other 200 sets of data are collected from the other half-moon. We randomly take 100 sets of data from each half-moon as samples, where 1 sample is labeled and the other samples are unlabelled. The remaining 200 sets of data in the Two Moons dataset are taken as the test data (i.e., new coming data in Section 5).

Figure 2 shows the experimental results of the proposed MKDPD algorithm and MR algorithm. In Figures 2(a) and

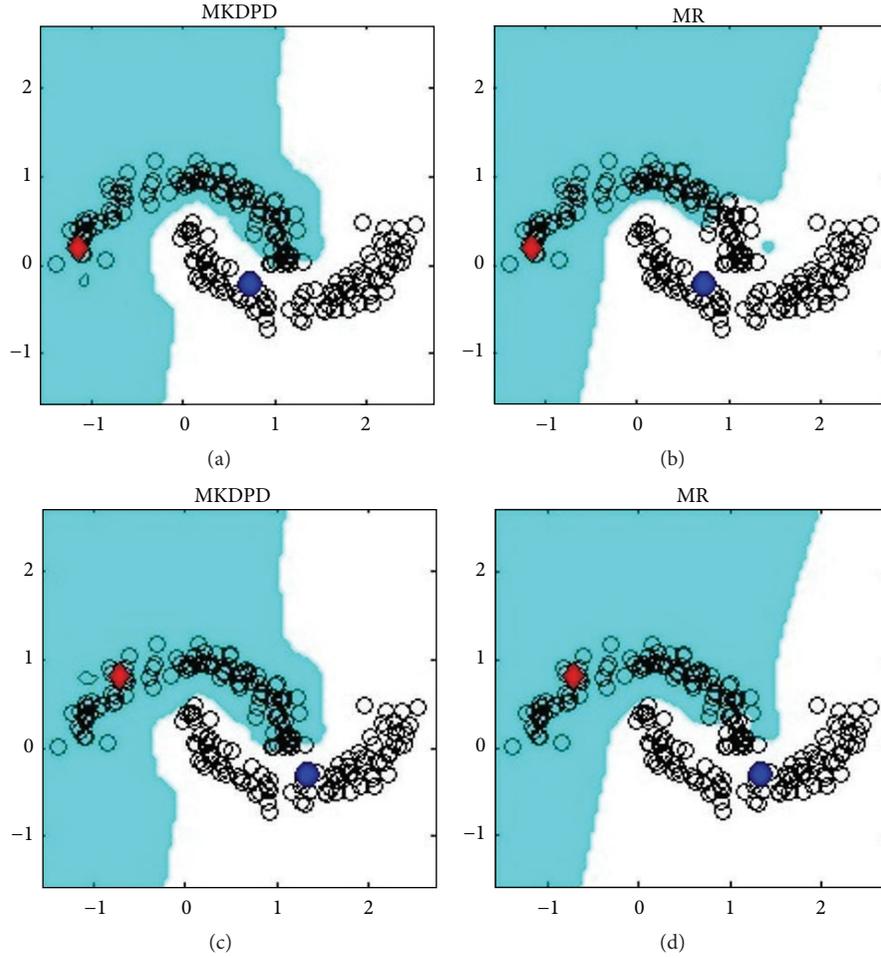


FIGURE 2: Two Moons Dataset: separating planes of MKDPD (see (a) and (c)) and MR (see (b) and (d)); the red and blue color points are labeled samples; the rest of the points are unlabeled samples.

2(b), although the labeled sample in the upper half-moon is badly located, the proposed MKDPD algorithm can still separate the upper half-moon from the lower half-moon, while MR algorithm fails at the one corner of the upper half-moon. In Figures 2(c) and 2(d), the labeled sample in the upper half-moon is located much near the center of the upper half-moon and accordingly the performance of MR algorithm becomes much better. However, the proposed MKDPD algorithm still outperforms MR algorithm in this circumstance.

The labeled samples are randomly taken for N times and each time will produce an experimental result. The average of experimental results is listed in Table 2. It can be seen from Table 2 that the proposed MKDPD algorithm outperforms MR algorithm under all circumstances.

There are two regularization terms in the proposed MKDPD algorithm and MR algorithm: manifold regularization $\lambda \|f\|_M^2$ and function regularization $\kappa \|f\|_H^2$, where $H = H_D$ in MKDPD or $H = H_X$ in MR. We keep λ unchanged, while letting κ change from 0 to 0.2 and comparing their performances (see Figure 3). As shown in Figure 3, the error rates of the proposed MKDPD algorithm and MR algorithm

TABLE 2: Average error rates (%) of MKDPD and MR on unlabeled samples of two moons set with different number of tests.

| Algorithm | Train times | | | |
|-----------|-------------|----------|----------|----------|
| | $N = 30$ | $N = 50$ | $N = 70$ | $N = 90$ |
| MKDPD | 0 | 0 | 0 | 0.0112 |
| MR | 0.8586 | 0.9293 | 0.7720 | 1.0325 |

decrease where the parameter κ decreases, but the error rate of the proposed MKDPD decreases much faster than that of MR algorithm.

6.3. Recognition of Handwritten Digits

6.3.1. USPS and MNIST Datasets. USPS (United States Postal) Dataset (<http://www.cs.nyu.edu/~roweis/data.html>) is a very popular dataset of handwritten digits, which contains 10 handwritten digits from “0” to “9”; each digit has 1100 images and each image is sized as 16×16 and can be converted into a 256-dimensional vector. We take the first 400 images of

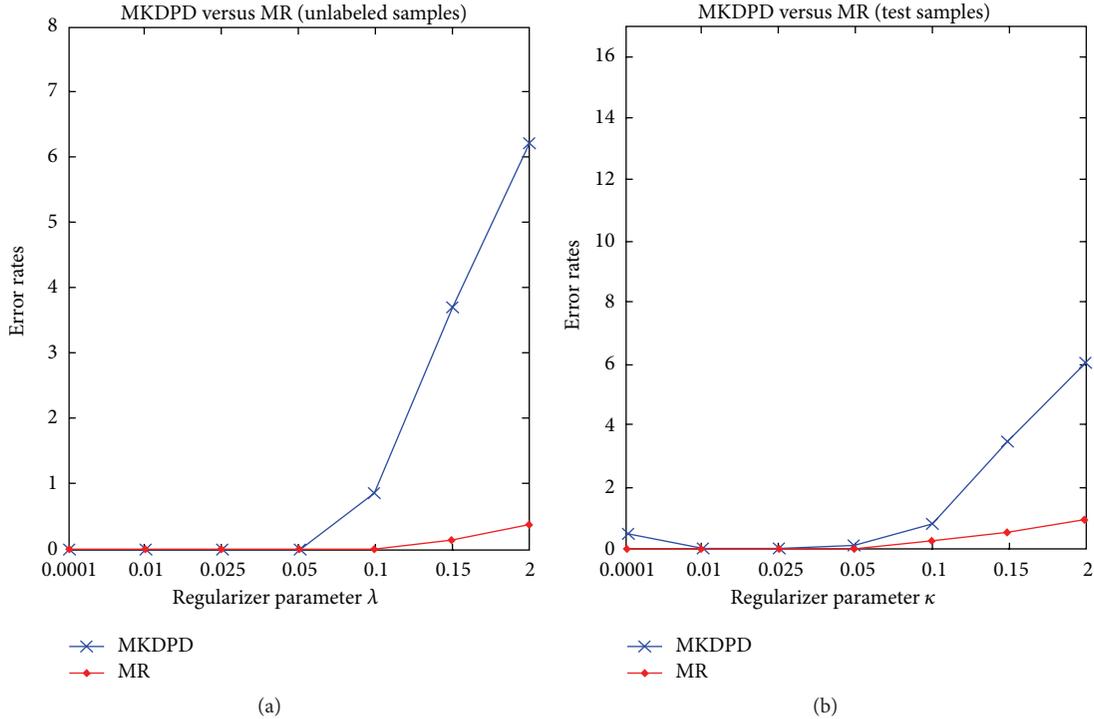


FIGURE 3: Error rates (%) of MKDPD and MR on unlabeled and testing samples of two moons set with the regular parameter κ varied from 0 to 0.2.

each digit as the samples and the remaining images as the test data (new coming data).

The MNIST (<http://yann.lecun.com/exdb/mnist/>) is another popular handwritten digits dataset, which contains a training set of 60000 images and a test set of 10000 images. Each image in MNIST is of size 28×28 and can be converted into a 784-dimensional vector. We select 400 sets of data from the training set as the samples for each digit and all data in the test set as the test data (new coming data).

6.3.2. The Two-Class Experiments. We randomly select two different digits from the ten digits to construct a binary classification problem, and then there is a total of 45 classification problems. For each binary classification problem, a sample is taken as the labeled sample for each digit and the remaining samples are taken as the unlabeled samples. To avoid the randomness, the labeled samples are randomly selected for ten times and each time will produce an experimental result. The average of ten experimental results is presented as the final experiment result.

The experimental results of the proposed MKDPD algorithm and MR algorithm are presented in Figure 4. In Figures 4(a), 4(b), 4(c), and 4(d), the x -axis represents the 45 binary classification problems, and the y -axis represents the error rates. The error rates on the unlabeled samples are shown in Figures 4(a) and 4(c), and the error rates on the test data are shown in Figures 4(b) and 4(d). As can be seen, the error rates of the proposed MKDPD algorithm are lower than those of the MR algorithm. Furthermore, the averages of the results of 45 binary classifications are listed in Table 3. It can

TABLE 3: Average error rates (%) of the two-class experiments on USPS, MNIST, and ISOLET datasets.

| Dataset | Algorithm | Unlabeled samples | Testing samples |
|---------|-----------|-------------------|-----------------|
| USPS | MKDPD | 2.5494 | 2.4723 |
| | MR | 3.1541 | 3.5774 |
| MNIST | MKDPD | 5.4213 | 6.0654 |
| | MR | 6.8557 | 8.1182 |
| ISOLET | MKDPD | 15.4067 | 17.7037 |
| | MR | 17.6039 | 24.9947 |

be seen from Table 3 that the proposed MKDPD algorithm outperforms the MR algorithm.

In Figures 4(e), 4(f), 4(g), and 4(h), the x -axis represents the error rates of labeling the unlabeled samples and the y -axis represents the error rates of labeling the test data. For a good learning algorithm, its error rates on the unlabeled samples and on the test data should be close to each other; that is, the scatter points in Figures 4(e), 4(f), 4(g), and 4(h) should be close to the diagonal line. Again, in this respect, the proposed MKDPD algorithm performs the MR algorithm better.

6.3.3. The Multiclass Experiments. In the multiclass experiments, there are 10 classes and each class corresponds to a digit. The labeled samples of each digit are randomly selected from its samples for 10 times and each time will produce an experimental result. The average of 10 experimental results is taken as the final result and listed in the first and second

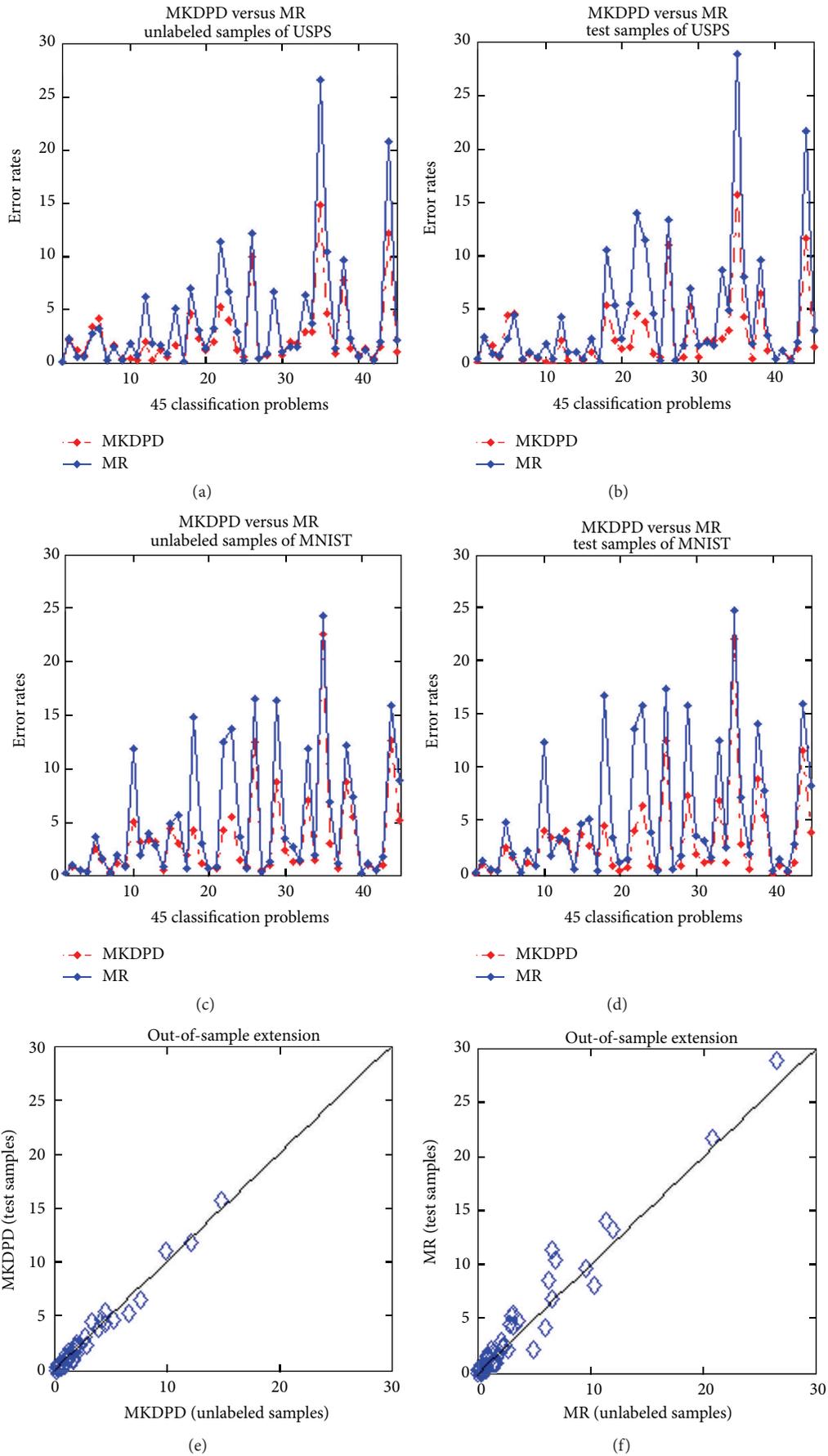


FIGURE 4: Continued.

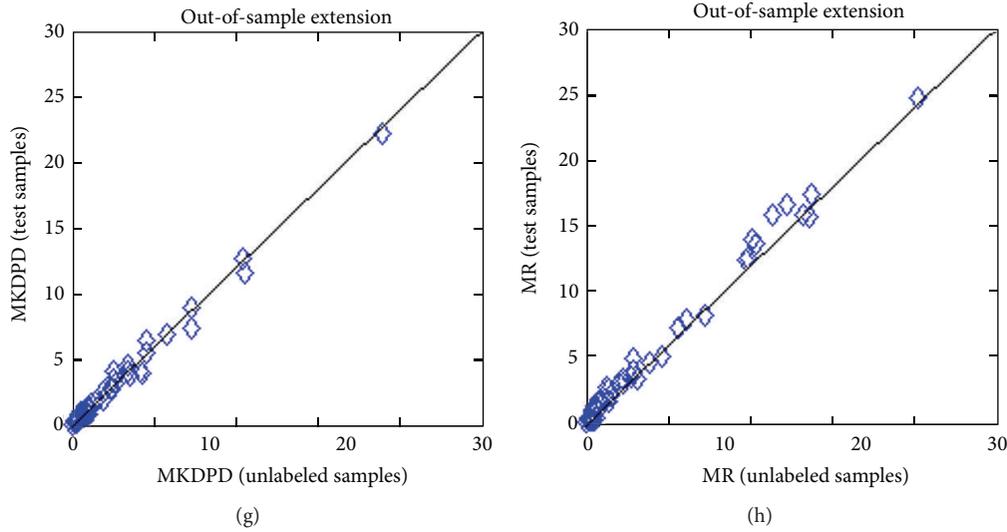


FIGURE 4: Two-class problem experiments: in (a, b, c, and d), (a) and (b) report the error rates (%) of MKDPD and MR on unlabeled and test samples of the USPS set, respectively, and (a) and (b) report the error rates (%) of MKDPD and MR on unlabeled and test samples of the MNIST set, respectively, where the y -axis is the 45 binary classification problems; in (e, f, g, and h), we report the scatter plots of error rate on unlabeled samples versus test samples. (e) and (g) are the scatter plots of error rates of MKDPD on USPS and MNIST, respectively, and (f) and (h) are the scatter plots of error rates of MR on USPS and MNIST, respectively.

column of Table 5. The number of labeled samples is set to be 1, 3, and 5, respectively. It can be seen from Table 5 that the proposed MKDPD algorithm outperforms the MR algorithm.

6.4. Recognition of Spoken Letters

6.4.1. ISOLET Dataset. ISOLET is a dataset of spoken letters and can be downloaded from UCI machine learning repository. ISOLET contains the utterances of 150 speakers who spoke 26 English letters twice, and then each speaker has 52 utterances. In the experiment, two subsets of ISOLET, whose names are ISOLET1 and ISOLET5 respectively, are directly download from (http://manifold.cs.uchicago.edu/manifold_regularization/manifold.html). Each subset contains the utterances of 30 speakers. We take the data in ISOLET1 as the samples and the data in ISOLET5 as the test data (new coming data).

6.4.2. The Two-Class Experiments. In the two-class experiments, the utterances of the first 13 English letters are classified as one class and the utterances of the last 13 English letters are classified as another class. We take the 52 utterances of one speaker from ISOLET1 as the labels samples and the utterances of the other speakers in ISOLET1 as the unlabeled samples. Since there are 30 speakers in ISOLET1, we can construct 30 two-class experiments this way and the 30 experimental results are presented in Figure 5. As can be seen from Figure 5, the proposed MKDPD algorithm outperforms the MR algorithm. The averages of 30 experimental results are listed in Table 4, which also shows that the proposed MKDPD performs better than the MR algorithm.

6.4.3. The Multiclass Experiments. In the multiclass experiments, the utterances of each English letter are classified as a class, and then there are 26 classes. We randomly select N speakers from ISOLET1 and take their utterances as the labeled samples. The utterances of the other speakers in ISOLET1 are then taken as the unlabeled samples. In order to alleviate the effect of randomness, the selection of N speakers is performed for 10 times and each time will produce an experimental result. The averages of 10 experimental results are taken as the final results and listed in the third column of Table 5. It can be observed from Table 5 that the proposed MKDPD algorithm achieves about 1%~3% improvements over the MR algorithm.

6.5. Face Recognition

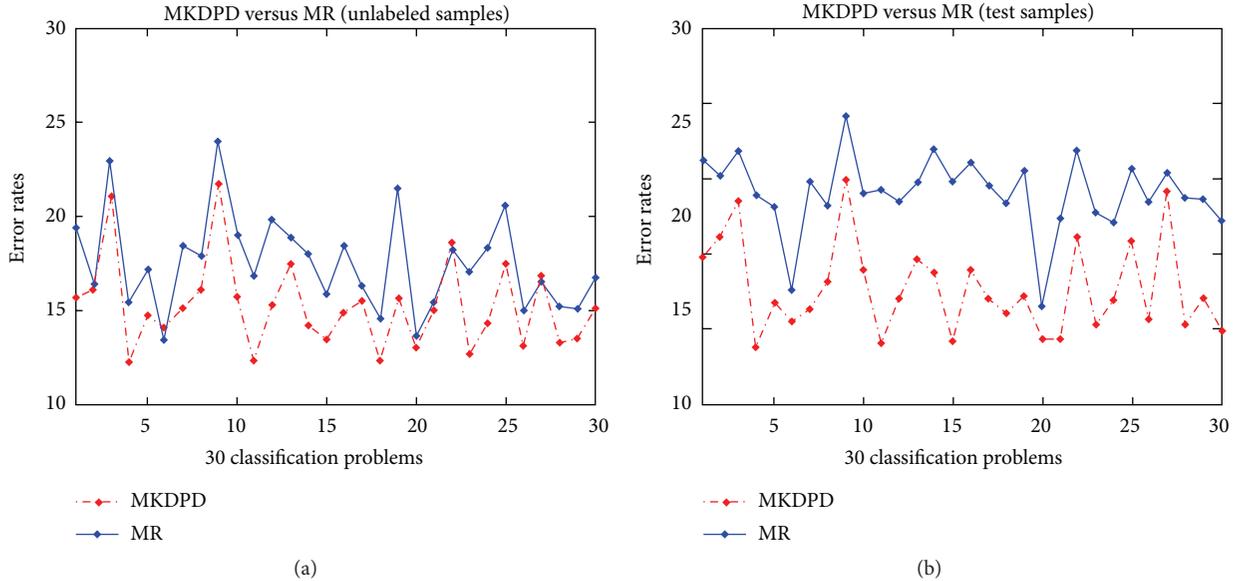
6.5.1. YALE-B and CMU-PIE Datasets. YALE-B (<http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html>) is a dataset of face images which consists of the face images of 10 persons; each person is photographed under 9 poses and 64 illuminations. In the experiment, we select the face images of 8 persons of 64 illuminations as the experimental dataset. Each image is cropped and resized to an image of 32×32 pixels. For each person, 50% of his face images are taken as the samples, while his other face images are taken as the test data (new coming data). CMU-PIE [38] is another dataset of face images which consists of more than 40000 images of 68 persons; each person is photographed under 4 expressions, 24 illuminations, and 13 poses. In the experiment, we select the face images of 8 persons of 3 poses and 24 illuminations as the experimental dataset. Each image is cropped and resized to an image of 32×32 pixels. Again, for each person, 50% of

TABLE 4: Average error rates (%) of the two-class experiments on Yale-B and CMU-PIE datasets with different number of labeled samples.

| Dataset | Algorithm | 3 labeled samples | | 5 labeled samples | | 7 labeled samples | |
|---------|-----------|-------------------|----------------|-------------------|----------------|-------------------|----------------|
| | | Unlabeled | Test | Unlabeled | Test | Unlabeled | Test |
| Yale-B | MKDPD | 35.2279 | 31.4955 | 30.4664 | 25.1786 | 27.3871 | 20.7087 |
| | MR | 36.7658 | 32.7455 | 31.7899 | 26.8136 | 28.8743 | 22.7087 |
| CMU-PIE | MKDPD | 23.9889 | 20.4167 | 13.6962 | 10.1042 | 8.5904 | 5.5754 |
| | MR | 26.0776 | 22.9712 | 15.9315 | 12.3611 | 10.0515 | 6.8452 |

TABLE 5: Error rates (%) of the multiclass problem experiments on five datasets: USPS, MNIST, ISOLET, Yale-B, and CMU-PIE, where the value in bracket is the number of labeled sample in each class.

| Dataset | Algorithm | Unlabeled (labeled samples) | | | | Test (labeled samples) | |
|---------|-----------|-----------------------------|--------------------|--------------------|--------------------|------------------------|--------------------|
| | | | | | | | |
| USPS | MKDPD | 19.8241 (1) | 12.7056 (3) | 10.3949 (5) | 18.1100 (1) | 10.9359 (3) | 8.5658 (5) |
| | MR | 23.5377 (1) | 14.5279 (3) | 11.2359 (5) | 21.2428 (1) | 13.3242 (3) | 10.2856 (5) |
| MNIST | MKDPD | 31.8342 (1) | 20.5939 (3) | 16.0872 (5) | 38.5650 (1) | 27.2950 (3) | 23.0300 (5) |
| | MR | 35.1206 (1) | 21.2741 (3) | 17.9026 (5) | 44.6050 (1) | 31.3800 (3) | 26.7300 (5) |
| ISOLET | MKDPD | 18.5743 (2) | 12.7060 (4) | 10.3704 (6) | 29.9469 (2) | 24.4297 (4) | 21.7241 (6) |
| | MR | 20.6300 (2) | 15.1305 (4) | 13.0057 (6) | 30.1790 (2) | 25.4111 (4) | 22.4801 (6) |
| Yale-B | MKDPD | 36.4440 (3) | 26.6991 (5) | 18.4750 (7) | 35.3516 (3) | 26.8203 (5) | 17.2070 (7) |
| | MR | 45.5172 (3) | 36.9676 (5) | 31.3500 (7) | 41.9531 (3) | 36.1406 (5) | 25.5078 (7) |
| CMU-PIE | MKDPD | 23.9773 (3) | 11.5591 (5) | 5.6034 (7) | 20.1968 (3) | 9.3056 (5) | 4.1551 (7) |
| | MR | 25.2904 (3) | 12.2715 (5) | 6.3793 (7) | 21.0532 (3) | 11.0069 (5) | 5.7292 (7) |

FIGURE 5: Two-class problem experiments: (a) is the error rates (%) of MKDPD and MR on unlabeled samples of ISOLET set, where the y -axis is the 30 binary classification problems; (b) is the error rates of MKDPD and MR on test samples of ISOLET set.

his face images are taken as the samples, while his other face images are taken as the test data (new coming data).

6.5.2. The Two-Class Experiments. We select the face images of two persons to construct a binary classification problem and then there are a total of 28 binary classification problems. For each binary classification problem, N samples of each person are taken as the labeled samples and the remaining samples are taken as the unlabeled samples. To avoid the

randomness, the N samples are randomly taken for 10 times and each time will produce an experimental result. The average of 10 experimental results is presented as the final results (see Figures 6 and 7), where the number of labeled samples is set to be $N = 3, 5,$ and $7,$ respectively. The average of 28 binary classification results is listed in Table 4. It can be seen that, compared with the MR algorithm, the proposed MKDPD algorithm achieves about 1%~5% improvements.

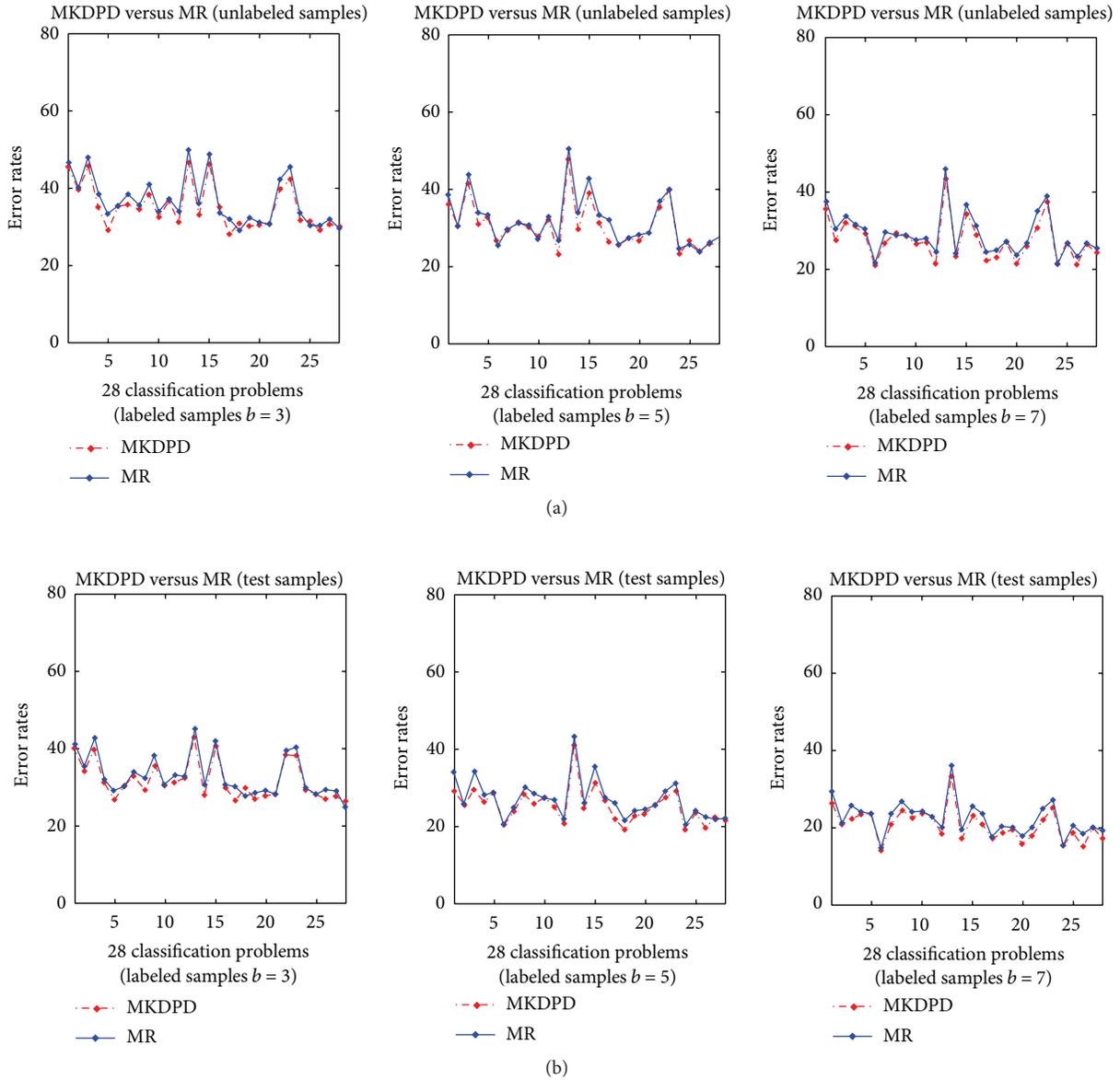


FIGURE 6: Two-class problem experiments: (a) is the error rates (%) of MKDPD and MR on unlabeled samples of the Yale-B set; (b) is the error rates (%) of MKDPD and MR on test samples of the Yale-B set. And the y -axis of each figure is the 28 binary classification problems and the value in bracket is the number of labeled samples in each class.

6.5.3. *The Multiclass Experiments.* In the multiclassification experiments, we take the face images of a person as one class and then there are 8 classes. N samples of each person are taken as the labeled samples and the remaining samples as the unlabeled samples. Again, the N labeled samples are randomly taken for 10 times and each time will produce an experimental result. The average of 10 experimental results is presented as the final result and listed in the last four columns of Table 5, where the number of labeled samples is set to be $N = 3, 5,$ and $7,$ respectively. As can be seen, the proposed MKDPD algorithm achieves about 1%~12% improvements to the MR algorithm.

7. Conclusion

In machine learning, one variable of a kernel function is often anchored on each given sample and thus derives a number of basic functions of the label function. The weights of basic functions in the label function are then trained by exploiting the labels of labeled samples. The basic functions derived this way are the same in shape, only different in the positions of data space. Obviously, these basic functions seem too simple to adapt to the changes of data distribution. For example, if the given samples are distributed unevenly, then in the area where there are too many samples are given, there

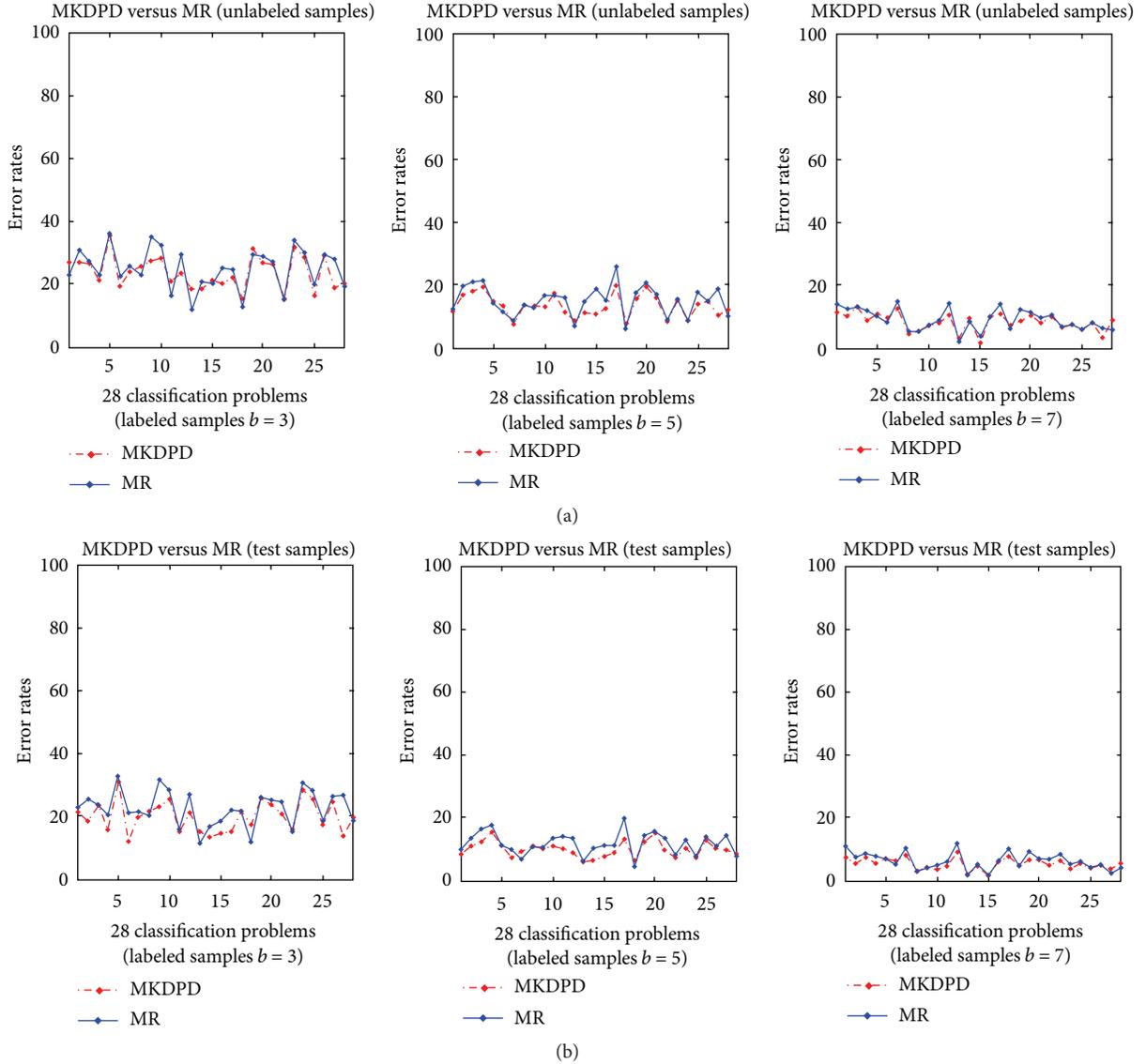


FIGURE 7: Two-class problem experiments: (a) is the error rates (%) of MKDPD and MR on unlabeled samples of the CMU-PIE set; (b) is the error rates (%) of MKDPD and MR on test samples of the CMU-PIE set. And the y -axis of each figure is the 28 binary classification problems and the value in bracket is the number of labeled sample in each class.

will be too many basic functions located in this area and maybe overlapped too much, while in the area where there are too few samples given, there will be too few basic functions located in this region and maybe overlapped too little or not overlapped at all.

In the MKDPD algorithm proposed in this paper, we adjust the basic functions according to the probabilities of data on the given samples. If the probability of data on a sample is large, then the number of samples in the vicinity of the sample will be large too and we can reduce the support of the basic function located on the sample accordingly to avoid overlapping too much with other basic functions. Likewise, if the probability of data on a sample is small, the number of samples in the vicinity of the sample will be small too and we can expand the support of the basic function located on the

sample accordingly to avoid overlapping too little with other basic functions. The experimental results justify the proposed MKDPD algorithm.

From the perspective of the applications of data classification, the aim of machine learning is to label the new coming data. Usually, there are three methods: unlearning, relearning, and semilearning. In the MKDPD algorithm proposed in this paper, there are two learning processes: learning the basic functions and learning the weights of basic functions. In this paper we propose a semilearning method based on the MKDPD algorithm. The proposed semilearning method regards the new coming data as unlabeled samples and mixes them with the original samples to relearn the basic functions, but still the original weights to combine the new basic functions to label the new coming data.

The proposed MKDPD-based method for labeling new coming data achieves a better trade-off between the computational accuracy and efficiency.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is supported in part by the Guangdong Provincial Science and Technology Major Projects of China under Grant 67000-42020009.

References

- [1] E. Parzen, "On estimation of a probability density function and mode," *Annals of Mathematical Statistics*, vol. 33, pp. 1065–1076, 1962.
- [2] S. Kay, "Model-based probability density function estimation," *IEEE Signal Processing Letters*, vol. 5, no. 12, pp. 318–320, 1998.
- [3] A. G. Bors and N. Nasios, "Kernel bandwidth estimation in methods based on probability density function modelling," in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR '08)*, Tampa, Fla, USA, December 2008.
- [4] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.
- [5] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley-Interscience, New York, NY, USA, 2004.
- [6] B. Scholkopf, A. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Journal of Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [7] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Müller, "Fisher discriminant analysis with kernels," in *Proceedings of the IEEE Signal Processing Society Workshop Neural Networks for Signal Processing IX*, pp. 41–48, Madison, Wis, USA, August 1999.
- [8] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: a geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.
- [9] S. Melacci and M. Belkin, "Laplacian support vector machines trained in the primal," *Journal of Machine Learning Research*, vol. 12, pp. 1149–1184, 2011.
- [10] B. Geng, D. Tao, C. Xu, L. J. Yang, and X.-S. Hua, "Ensemble manifold regularization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 6, pp. 1227–1233, 2012.
- [11] Y. Luo, D. C. Tao, B. Geng, C. Xu, and S. J. Maybank, "Manifold regularized multitask learning for semi-supervised multilabel image classification," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 523–536, 2013.
- [12] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.
- [13] S. S. Bucak, R. Jin, and A. K. Jain, "Multiple kernel learning for visual object recognition: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1354–1369, 2014.
- [14] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.
- [15] F. A. Tobar, S.-Y. Kung, and D. P. Mandic, "Multikernel least mean square algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 265–277, 2014.
- [16] Y.-L. Xu, D.-R. Chen, H.-X. Li, and L. Liu, "Least square regularized regression in sum space," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 4, pp. 635–646, 2013.
- [17] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge, UK, 2004.
- [18] X. J. Zhu, "Semi-supervised learning literature survey," *Computer Sciences TR 1530*, 2008.
- [19] 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.
- [20] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: from transductive to semi-supervised learning," in *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, pp. 825–832, August 2005.
- [21] X. J. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th International Conference on Machine Learning (ICML '03)*, Washington, DC, USA, 2003.
- [22] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems 16*, 2004.
- [23] F. P. Nie, D. Xu, I. W. Tsang, and C. S. Zhang, "Flexible manifold embedding: a framework for semi-supervised and unsupervised dimension reduction," *IEEE Transactions on Image Processing*, vol. 19, no. 7, pp. 1921–1932, 2010.
- [24] M. Y. Fan, N. N. Gu, H. Qiao, and B. Zhang, "Sparse regularization for semi-supervised classification," *Pattern Recognition*, vol. 44, no. 8, pp. 1777–1784, 2011.
- [25] J.-W. Xu, A. R. C. Paiva, I. Park, and J. C. Principe, "A reproducing kernel Hilbert space framework for information-theoretic learning," *IEEE Transactions on Signal Processing*, vol. 56, no. 12, pp. 5891–5902, 2008.
- [26] S. Liwicki, S. Zafeiriou, G. Tzimiropoulos, and M. Pantic, "Efficient online subspace learning with an indefinite kernel for visual tracking and recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 10, pp. 1624–1636, 2012.
- [27] K. Slavakis, P. Bouboulis, and S. Theodoridis, "Adaptive multi-regression in reproducing kernel hilbert spaces: the multiaccess MIMO channel case," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 2, pp. 260–276, 2012.
- [28] G. Q. Li, C. Y. Wen, Z. G. Li, A. M. Zhang, F. Yang, and K. Z. Mao, "Model-based online learning with kernels," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 3, pp. 356–369, 2013.
- [29] P. Gurrām and H. Kwon, "Contextual SVM using Hilbert space embedding for hyperspectral classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 5, pp. 1031–1035, 2013.
- [30] Y. Gu, S. Wang, and X. Jia, "Spectral unmixing in multiple-kernel hilbert space for hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 7, pp. 3968–3981, 2013.

- [31] B. Scholkopf and A. J. Smola, *Learning with Kernels*, MIT Press, Cambridge, Mass, USA, 2001.
- [32] S. Wu and S.-I. Amari, "Conformal transformation of kernel functions: a data-dependent way to improve support vector machine classifiers," *Neural Processing Letters*, vol. 15, no. 1, pp. 59–67, 2002.
- [33] P. Gurram and H. Kwon, "Sparse kernel-based ensemble learning with fully optimized kernel parameters for hyperspectral classification problems," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 2, pp. 787–802, 2013.
- [34] J. Huang, P. C. Yuen, W.-S. Chen, and J. H. Lai, "Choosing parameters of kernel subspace LDA for recognition of face images under pose and illumination variations," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 37, no. 4, pp. 847–862, 2007.
- [35] G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.
- [36] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.
- [37] J. J. Thiagarajan, K. N. Ramamurthy, and A. Spanias, "Multiple kernel sparse representations for supervised and unsupervised learning," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 2905–2915, 2014.
- [38] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1615–1618, 2003.

Research Article

Hand Gesture Recognition Using Particle Swarm Movement

Clementine Nyirarugira, Hyo-rim Choi, and TaeYong Kim

Graduate School of Advanced Imaging Sciences, Film, and Multimedia, Chung-Ang University, Seoul 156-756, Republic of Korea

Correspondence should be addressed to TaeYong Kim; kimty@cau.ac.kr

Received 4 March 2016; Revised 11 May 2016; Accepted 24 May 2016

Academic Editor: Erik Cuevas

Copyright © 2016 Clementine Nyirarugira 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 present a gesture recognition method derived from particle swarm movement for free-air hand gesture recognition. Online gesture recognition remains a difficult problem due to uncertainty in vision-based gesture boundary detection methods. We suggest an automated process of segmenting meaningful gesture trajectories based on particle swarm movement. A subgesture detection and reasoning method is incorporated in the proposed recognizer to avoid premature gesture spotting. Evaluation of the proposed method shows promising recognition results: 97.6% on preisolated gestures, 94.9% on stream gestures with assistive boundary indicators, and 94.2% for blind gesture spotting on digit gesture vocabulary. The proposed recognizer requires fewer computation resources; thus it is a good candidate for real-time applications.

1. Introduction

Hand gestures are a powerful human to human communication channel that forms a major part of information transfer in our daily life. Incorporating hand gestures into the human computer interface is becoming an important research area. The usage of vision-based hand gesture is often preferred for its noncumbersome interaction [1, 2]. To interact with a device using hands, computers should be able to visually detect the hand and recognize its gestures from video input [3–5].

The latest computer vision technologies make real-time hand detection and gesture recognition promising [6]. Many different approaches that use hand as an interface device have been proposed [1, 2]. To support gestures interaction, a recognizer must be integrated into the system and trained to the specific gestures the system will support. However, most of recognizers have inherent limitations in the types of gestures they can efficiently discriminate [7], which often results from high intergesture categories correlation.

In addition, online hand gesture recognition brings more challenges, as gestures are often characterized by unpredictable boundary noise (see Figure 1(a)), due to the lack of perfect vision-based gesture segmentation methods, between gestures (when gestures are coarticulated) (see Figure 1(b)), leading to ambiguous recognition [8].

In online hand gesture recognition systems, often two assumptions are made: the existence of gesture boundary indicators or blind gesture spotting, the latter being more difficult. Each approach has different implications and the preference of one over the other is application-dependent. Few attempts have been made in blind spotting [9, 10] that are predominant in real-world applications. Blind gesture recognition makes the classification ambiguous and more difficult, as less information is presented to the recognizer.

One of the challenges is that gestures vocabularies often contain highly correlated gestures that lead to ambiguous recognition [8, 11]. For instance, in the digit gesture vocabulary the hand motion performed to gesticulate the digit two is a part of the one performed for the digit three. To avoid ambiguity in recognition, additional actions are often required. Alon et al. [10] proposed a template matching method with subgesture detection and reasoning method, which avoid premature gesture spotting. In [9], the gestures boundaries are detected prior to gesture recognition using Hidden Markov Models (HMM). These methods produce promising recognition rates; however, the computation resources are still high.

In this paper, we adapt particle swarm optimization (PSO) to the problem of gesture recognition. PSO is a pattern search method [12]. In general, a PSO algorithm is initialized with a group of N particles. Each particle is characterized

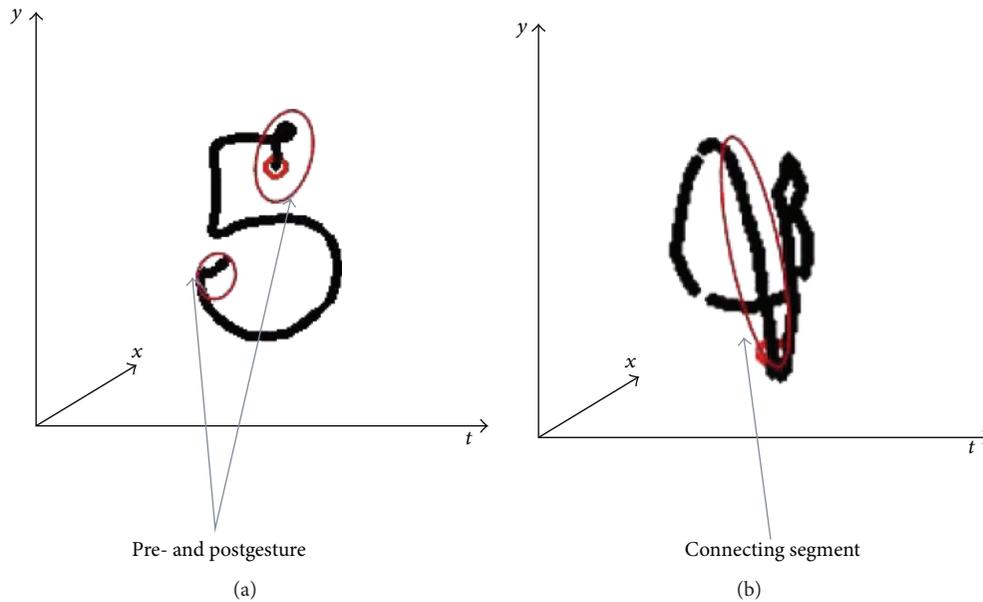


FIGURE 1: Distorted gestures: (a) boundary noise that may lead to classify 5 as 8 and (b) connecting segment that lead to classify 4 as 9.

by its personal best position, which is updated according to its fitness value (likelihood). Within the gesture recognition context, the search/solution space is composed of gesture templates to which we assign particles and allow them to evolve through a deterministic matching process, guided by observed data. The recognized gesture category is the one whose particle has the highest matching score and has reached or is closest to the end of the template. Gesture detection under PSO matching offers a more efficient way of gesture segmentation as gesture boundaries can be inferred directly from the matching process, that is, the particle's personal best positions; thus, there is no need for backtracking. The contribution of this paper can be summarized as follows:

- (1) An automated process of segmenting meaningful gesture trajectories based on particle swarm movement is proposed.
- (2) To avoid premature gesture spotting, a subgesture detection and reasoning method is incorporated within the proposed recognizer.
- (3) Reduce considerably the processing time of gesture spotting, as gesture boundaries can be inferred directly from particle's personal best position.

We evaluate the performance of the proposed method under three gesture spotting and recognition assumptions, that is, manually spotted (with consistent boundaries), pre-segmented with boundary indicators (with possible boundary noise), and blind gesture spotting [8]. The remainder of this paper is organized as follows; Section 2 discusses recent related works in dynamic hand gesture recognition. Section 3 gives details on the proposed method, followed by Section 4 that discusses a series of experiments to evaluate the proposed methods. Finally, Section 5 concludes this work.

2. Hand Gesture Recognition

Hand gesture recognition is a difficult and challenging problem that has been addressed in many ways. A widely used approach is the Hidden Markov Model (HMM) [9, 13–16]. HMM based gesture recognition methods represent each gesture by a set of states associated with probabilities (initial, transition, and observation) learned from the training examples. HMM recognizers choose a model with the best likelihood and classify a given gesture to the corresponding gesture category. Although HMM recognition systems choose a model with the best likelihood, it is not guaranteed that the pattern is really similar to the reference gesture unless the likelihood value is high enough, above some threshold. In the case a simple threshold does not work well, a sophisticated threshold model can be derived as done in [14] or other verification mechanisms are applied as done in [4].

To produce good results, HMM need to be well trained to get good representative models [9]. Rule based trajectory segmentation for modeling the hand motion trajectory has been proposed [13], to provide a robust initialization. The authors did an extensive study on good initialization of the HMM, which is often of primary concern in HMM based recognition method. They derived an automated process for determining the number of states and a robust initialization of HMM. The proposed method can separate each angular state of the training data at the initialization step, thus providing a solution to mitigate the ambiguities on initializing the HMM and increase the recognition of the HMM.

An automatic system that handles hand gesture spotting and recognition simultaneously based on a generative model as HMM has been proposed in [9]. To spot meaningful (key) gestures of numbers (0–9) accurately, a stochastic method for designing a nongesture model with HMM was proposed

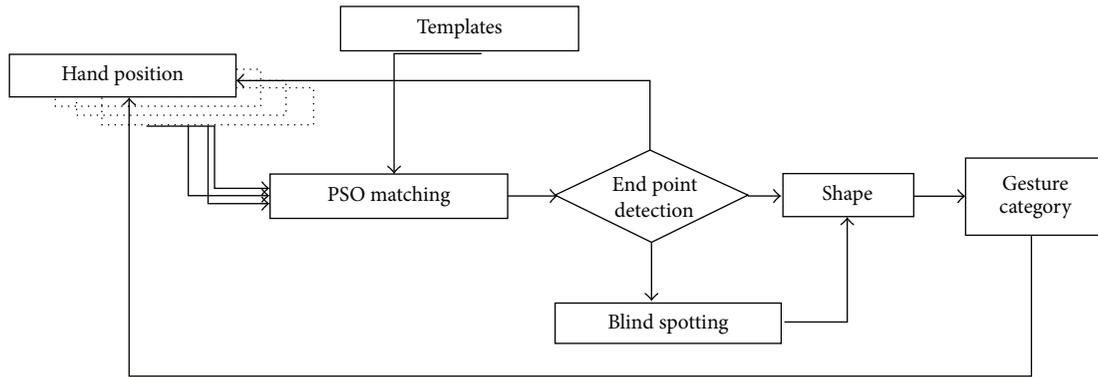


FIGURE 2: Flowchart of the proposed recognizer.

without training data. The nongesture model provides a confidence measure that is used as an adaptive threshold to find the start and the end point of meaningful gestures, which are embedded in the input video stream. Threshold model approach is often computationally expensive, as one has to create a big size nongesture model [4]. To filter out garbage gesture, in [4], authors appeal to a simple Gaussian model threshold based on a single Gaussian probability density, learned during the training process. The main disadvantages of HMM based recognition methods are that they require a large number of samples and long training time to calibrate the models [14]. When there are not enough training examples, template matching methods such as Dynamic Time Warping (DTW) are preferred.

DTW is another approach often used for dynamic gesture recognition task [3, 17, 18]. The DTW based recognition methods attempt to line up a given sequence to gestures templates. To produce good results, many templates may be required to take into account variation of a given gesture category as it may be the case for other template matching methods. Moreover, DTW is shown, in time series, to be sensitive to noise and outliers. Different distance metrics have been derived to improve DTW results. Probability-based DTW and bag-of-visual-and-depth-words for human gesture recognition in RGB-D (Red Green Blue-Depth) used a soft distance based on the probabilistic similarity measure [17]. The aforementioned derived distance measure improves the recognition rate of classical DTW that uses Euclidean distance as a cost function. In [19], a comparative evaluation of six trajectory distance measures was performed. The longest common subsequence (LCS) measure outperforms the others on datasets with varying characteristics. The LCS is a string matching algorithm that focuses on the matched subsequence. This makes it robust to noise. The LCS has recently gained more attention and has been successfully used in dynamic hand gesture recognition systems.

Stratified hand gesture using normalized longest common subsequence with rough sets [8] aimed to increase LCS discrimination capabilities, as LCS is a global alignment method. The authors achieved good performance through the normalization of LCS and pairing the LCS with rough sets theory. Gestures are represented using rough set

approximations, through which discriminative information was generated and used to resolve ambiguous recognition.

The success of a gesture recognition system does not only rely on recognition but also should run in real time. Vision-based gesture segmentation is difficult and gesture boundary detection is often computationally expensive. Most approaches rely on the usage of sliding windows. However, without constraining users to perform gestures at nearly the same speed, window management is challenging which may result from high variation in gesture size and speed. The failure to predict the window size leads to a recomputation of the likelihood at every window size update, with delay. It also may lead to premature gesture recognition. To avoid uncertainty in the choice of the sliding windows, different approaches are followed such as the usage of gesture boundary indicators or classification of motion primitive. The aforementioned boundary indicators increase gesture detection rates; however, users have to bear them in mind and apply them for a successful interaction. This might be tiring based on application, such as string recognition system [8].

The following section introduces gestures detection using particle swarm movement, which does not require explicit gesture boundary detection, nor the usage of sliding windows, as gesture boundary detection is inferred from PSO matching process.

3. PSO Based Gesture Recognition

PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution. In general, a PSO algorithm is initialized with a group of N particles. Each particle is characterized by its personal best position, which is updated according to its fitness value (likelihood). The flowchart of the proposed recognizer is depicted in Figure 2.

3.1. Gesture Representation. Given a sequence of images (video file) in which a hand gesture is being performed, the gesturing hand is first detected and its trajectory p_s is acquired; $p^s = [p_1, p_2, \dots, p_t]$, with length t^s .

Each point is represented in 2D, that is, $p^s(t) = [x(t), y(t)]$.

- (1) We first computed the Euclidean distance from the start and end of the gesture trajectory to infer the shape of a gesture, in the sense of being either closed or open, as follows:

$$\text{Euclid}_p^s = d(p_n^s - p_1^s). \quad (1)$$

This will become handy in recognizing gestures with the same motion, but different shapes, such as digit 0 and digit 6.

- (2) Gesture trajectory is then mapped into the motion orientation segments:

$$\theta^s = [\theta_1, \theta_2, \dots, \theta_{t^s-1}], \quad (2)$$

where $\theta^s(t) = \text{atan2}(\Delta y(t), \Delta x(t))$. Orientation segments are created as detailed in [8].

A gesture is then represented by $[\theta^s, \text{Euclid}_p^s]$.

3.2. Gesture Recognition Using Particle Swarm Movement. PSO was originally developed for continuous problems. Adapting it to discrete domains is somehow tricky, approximations need to be made, and some concepts may lose some of their meaning. In our gesture recognition setting, the search space is composed of gesture templates, to which particles are assigned. The solution space is defined as follows:

$$\Psi = \{\theta^i = [\theta_1^i, \theta_2^i, \dots, \theta_{n_i}^i]\}_{i=1}^L, \quad (3)$$

where $L \leftarrow 1, \dots, |\Psi|$ is number of particles, θ^i is the i th template, and n_i is the length of the i th template.

Gestures recognition using particles swarm is performed as follows.

- (1) Initialize particles in the search (templates) space,

$$\{x_j^i = 0; p_{\text{best}}^i = x_j^i\}_{i=1}^L, \quad (4)$$

where x_j^i represent the j th position of the i th particle, in the i th template, and p_{best}^i is the personal best position of the i th particle, to simplify notation $x_j^i = j$ when used as an index.

- (2) Evaluate the particles' fitness function.

For each motion orientation segment θ_t , a matching score is calculated and the particle's fitness value is computed as follows:

$$\begin{aligned} \gamma &= f(\theta_j^i, \theta^t); \quad \gamma \in [-1, 0, 1]; \\ F_t(\theta^i) &= F_{t-1}(\theta^i) + |\theta_t| * f(\theta_j^i, \theta^t), \end{aligned} \quad (5)$$

where $\gamma = -1$ when $\theta_j^i \neq \theta^t$ and $p_{\text{best}}^i \neq 0$, $\gamma = 0$ when $\theta_j^i \neq \theta^t$, and $\gamma = 1$ when $\theta_j^i = \theta^t$.

- (3) The personal best position is updated only when the current fitness value is greater than the previous one:

$$\begin{aligned} p_{\text{best}}^i &= p_{\text{best}}^i + 1, \\ \text{if } F_t(\theta^i) &> F_{t-1}(\theta^i). \end{aligned} \quad (6)$$

In our gesture recognition setting each particle moves locally towards a more likely position in the search space:

$$x_j^i = x_j^i + 1. \quad (7)$$

- (4) Compute a subset of possible solution:

$$\begin{aligned} \psi_{\text{sol}} &= \{\theta^i\} \\ \text{such that } p_{\text{best}}^i &\geq \beta^i, \end{aligned} \quad (8)$$

where β^i is a threshold for the i th gesture category.

- (5) If $\Psi_{\text{sol}} \neq \Phi$, the recognized gesture category i_0 is determined as follows:

$$i_0 = \arg \max_{\theta^i \in \Psi_{\text{sol}}} \left(\frac{F_t(\theta^i)}{|\theta^i| + |\theta|} \right). \quad (9)$$

3.3. Blind Gestures Spotting through PSO Matching. Given a set of templates Ψ representing gesture categories within a gesture vocabulary, a list of subgestures-super gestures relationship is generated by cross matching all templates and populating their personal best position in the matrix G , $G \leftarrow T \times T$:

$$G_{n,n} = \begin{pmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,n} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n,1} & g_{n,2} & \cdots & g_{n,n} \end{pmatrix}, \quad (10)$$

where $G_{i,j}$ represents the personal best position of the particle j when matched with particle i , that is, the number of their common segments. A super gesture category is the one that has a part that overlaps with another whole gesture within the same gesture vocabulary. The list of subgesture or super gesture will be checked during gesture blind spotting to avoid premature gesture spotting. The sub_super gesture relationship is detected as follows:

$G \leftarrow T \times T$: created using PSO matching

for $i = 1: n$

for $j = 1: n$

if $(G_{i,j} = |T^i|)$

$Y_{(\text{sub}, \text{sup})} \leftarrow \langle i, j \rangle$

end if

end for

end for

TABLE 1: Confusion matrix of recognition of gesture with known boundaries.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|-------|------|-------|------|------|------|-------|------|-------|-------|
| 0 | 98.1% | | | | | | 1.9% | | | |
| 1 | | 100% | | | | | | | | |
| 2 | | | 93.3% | | | | | 5.3% | | |
| 3 | | | | 100% | | | | | | |
| 4 | | | | | 100% | | | | | |
| 5 | | | | | | 100% | | | | |
| 6 | 6.1% | | | | | | 94.9% | | | |
| 7 | | | | | | | | 100% | | |
| 8 | | | | | | | | | 92.3% | |
| 9 | | | | | | | | | | 97.1% |

Gesture spotting is the process of detecting a meaningful gesture in a continuous gesture stream. The following are gestures spotting rules used with particle swarm movement matching. Given a list of sub-super gesture relationship, $Y_{(sub,sup)}$, a gesture is detected as follows:

- (1) To spot a gesture, we first check if there is a particle that has already reached the end of its corresponding template:

$$i_0 \leftarrow p_{best}^i == |T^i|, \quad \forall i \in L. \quad (11)$$

- (2) Detection of nonsubgesture: if the detected gesture i_0 is not a subgesture, $i_0 \notin Y_{(sub)}$, that is, it could be a super gesture or normal gesture, then it is reported.
- (3) Spotting subgestures: if the detected gesture is subgesture, that is, $\langle i_{sub}^0, i_{sup}^0 \rangle$ relationship exists, and the following condition checks out, a subgesture is reported:

$$\Delta F \left(p_{best}^{i_{sup}^0} \right) == 0. \quad (12)$$

Otherwise, a delay is observed when (12) is not true, that is, waiting for the next segment to avoid premature gesture spotting.

Gesture spotting with PSO matching does not require a traceback as gesture boundaries (start and end points) can be directly inferred from the p_{best} .

4. Experimental Results

To evaluate the performance of the proposed method, we collected gestures video clips at 30 FPS using Kinect v1, from five people, drawing in the air digits “0–9.” There was no restriction made on gesturing speed or size. Implementation was done using C++ and OpenCV libraries. PSO algorithm is implemented based on a discrete version [20], which is modified to track gestures and to accommodate the blind gesture spotting, and parameters for PSO are tuned by the speed of a gesturing hand. In the first round, each video contains ten or more gestures with pause between gestures,

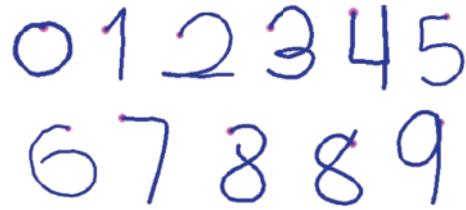


FIGURE 3: Gesture vocabulary.

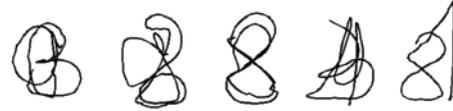


FIGURE 4: Coarticulated gestures (left to right): 93, 58, 88, 34, and 81.

mounting to 50 gestures for each gesture category as shown in Figure 3. In the second round, each video file contained more than 2 random coarticulated gestures with possible connecting segments; see Figure 4. The evaluation of the effectiveness of the recognition method was performed using the following expression:

$$\text{recognition_rate} = \left(1 - \left(\frac{I + D + S}{N} \right) \right) \times 100, \quad (13)$$

where I stand for the number of inserted gestures, D missed gestures, and S substitution errors. The computed rate shows the prediction of the right response when these gestures are used in control systems or gaming.

The first evaluation was done on manually spotted gestures. An average recognition rate of 97.5% was obtained on 524 gestures, as shown in Table 1. The recognition rate shows that template matching based on the particle’s personal best positions performs well.

In the recognition on stream gestures with assistive boundary indicators, we perform temporal gesture segmentation as a preprocessing step; depth profile is used as temporal segmentation indicator, as the hand pull-back at the end of a gesture. The start of a gesture is detected when a hand is extended and starts to move. Similarly the end of the gesture is detected when a hand is retracted. The resulting gestures may

TABLE 2: Confusion matrix recognition of gesture with unknown boundaries.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|-------|-------|-------|-------|-------|-------|-------|-----|-------|-------|
| 0 | 95.7% | | | | | | 4.3% | | | |
| 1 | | 92.6% | | | 6.8% | | | | | |
| 2 | | | 93.5% | | | | | | | |
| 3 | | | | 95.7% | | | | | | |
| 4 | | | | | 93.1% | | | | | 6.9% |
| 5 | | | | | | 97.9% | | | | |
| 6 | 4.2% | | | | | | 95.8% | | | |
| 7 | | | 2.9% | | | | | 96% | | |
| 8 | | | | | 3.6% | | | | 91.5% | |
| 9 | | | | | | | | | | 98.1% |

TABLE 3: Blind gesture spotting.

| | Single gestures | Paired gestures |
|--------------------|-----------------|-----------------|
| Number of gestures | 130 | 45 |
| Recognition rate | 94.2% | 86.36% |

contain additional parts, pre- and postgestures considered as boundary noise, since they are not a part of the gesture; see Figure 1. Table 2 shows the confusion matrix; the average recognition rate on stream digits is 94.9%. The recognition rate dropped by almost 3%. Due to boundary noise, digit one is confused with four, four with nine, and seven with two. This can be explained by the sub-super gesture relations existing between confused gestures and the presence of boundary noise.

We evaluate blind gesture spotting on continuous gestures, without gesture boundary indicator or special treatment of connecting segments. Each video file contains two or more random gestures. For paired gestures, a pair is recognized, only if both gestures are correctly recognized. As shown in Table 3, the recognition rate of single gesture shows good performance despite coarticulation with the others. However, the recognition rate on paired gestures drops as gestures are recognized together.

We compare the proposed recognizer to the state-of-the-art methods in dynamic hand gesture recognition, based on the recognition accuracy and the processing time, on the same gesture vocabulary. Note that the experimental environment may be different; thus the following comparison is to be understood as the ability of the methods under consideration to discriminate a given gesture vocabulary and relative processing time, in general.

Table 4 summarizes performance of some implementation of gesture recognizers. As shown in Table 4, in their experiments, the authors of [13] evaluated their method, which models lines and curves using Von Mises Distribution (VMD) and realigns them according to gesture structures, by comparing it to a fixed number of states. The method produced good recognition rate, 97.1%, on preisolated gestures.

In [9], a recognition rate of 98.3% was achieved on training samples and 97.7% on test samples, showing that a well-trained HMM produces better rate, as reported in

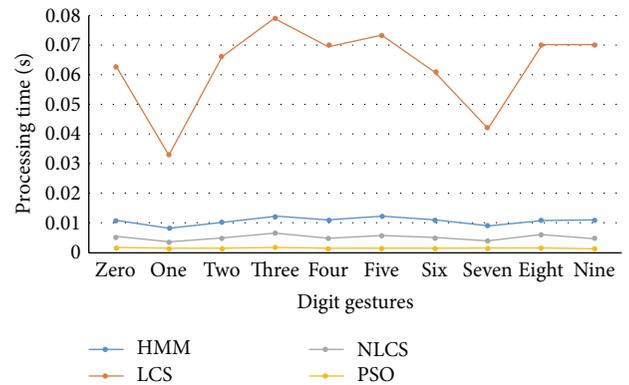


FIGURE 5: Processing time comparison.

different studies. The main contribution of [9] was to provide a gesture spotting mechanism, through the start and end points detection, under the HMM. A nongesture model was constructed by combining all HMM models of the gesture vocabulary and used as a threshold. They achieved a recognition rate of 93.3% on stream digits. The Most Probable Longest Common Subsequence (MPLCS) in [11] modeled different intragesture variations through the use of Mixture of Gaussian Models (GMM). Recognition rates of 98.7% and 94% were achieved on preisolated and stream gestures, respectively. In this work, we achieved on average, the recognition rates of 97.56% and 94.5%, on manually segmented and stream gestures with unknown boundaries, respectively.

Figure 5 shows comparative processing time of HMM, LCS, and PSO based gesture recognition on digit gesture vocabulary. Point matching LCS has the highest processing time. In our evaluation, the HMM requires approximately 16.8% of LCS, the normalized longest common subsequence (NLCS) with segment matching requires 8.1%, and the proposed PSO based matching method requires only 2.3% of the highest processing time.

5. Conclusion

An online gesture recognition method based on particle swarm optimization has been derived through this work.

TABLE 4: Dynamic gesture recognition with HMM, LCS, and PSO.

| Paper | Representation | Isolated | Stream | Vocabulary | Method |
|-----------------|----------------|----------|--------|--------------|--------|
| [13], 2014 | VMD | 97.1% | — | Alpha number | HMM |
| [9], 2014 | Codewords | 97.7% | 93.3% | Number | HMM |
| [8], 2015 | Codewords | 88.7% | — | Number | HMM |
| [11], 2013 | GMM | 98.7% | 94.0% | Number | LCS |
| This work, 2016 | Codewords | 97.5% | 94.2% | Number | PSO |

Gesture vocabularies often contain highly correlated gestures which reduce the accuracy of most recognizers. Furthermore, the choice of assistive temporal segmentation cues is challenging and their usage may not be the right approach in the systems that need a string input such as word search or recognition. In this paper, we integrated a subgesture reasoning mechanism into the proposed PSO based matching method, to avoid premature gesture spotting.

The proposed recognizer showed promising recognition rate under different assumptions: a recognition rate of 97.5% was obtained on gestures with known boundaries, 94.9% on gesture with boundary noise, and 94.2% on continuous gesture without assistive boundary detection. A pairwise gesture recognition evaluation achieved a rate of 86.3%. The processing time was reduced to 2.3% of LCS point matching method. In the comparison to existing gesture recognition method, the proposed recognizer shows similar recognition rate and requires less processing time.

In the future work, we plan a further analysis on paired gestures, as some intergesture categories relationship cannot be detected from single gesture analysis basis.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (NRF-2015R1D1A1A01058394) and by the Chung-Ang University Research Scholarship grants in 2015.

References

- [1] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Communications of the ACM*, vol. 54, no. 2, pp. 60–71, 2011.
- [2] S. Lian, W. Hu, and K. Wang, "Automatic user state recognition for hand gesture based low-cost television control system," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 1, pp. 107–115, 2014.
- [3] P. Doliotis, A. Stefan, C. McMurrough, D. Eckhard, and V. Athitsos, "Comparing gesture recognition accuracy using color and depth information," in *Proceedings of the 4th ACM International Conference on Pervasive Technologies Related to Assistive Environments (PETRA '11)*, May 2011.
- [4] J. H. Lee, T. Delbruck, M. Pfeiffer et al., "Real-time gesture interface based on event-driven processing from stereo silicon retinas," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 12, pp. 2250–2263, 2014.
- [5] C. Nyirarugira and T. Kim, "Adaptive differential evolution algorithm for real time object tracking," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 4, pp. 833–838, 2013.
- [6] M. B. Holte, T. B. Moeslund, and P. Fihl, "View-invariant gesture recognition using 3D optical flow and harmonic motion context," *Computer Vision and Image Understanding*, vol. 114, no. 12, pp. 1353–1361, 2010.
- [7] L. Anthony, R. D. Vatavu, and J. O. Wobbrock, "Understanding the consistency of users' pen and finger stroke gesture articulation," in *Proceedings of the Graphics Interface (GI '13)*, pp. 87–94, Regina, Canada, May 2013.
- [8] C. Nyirarugira and T. Kim, "Stratified gesture recognition using the normalized longest common subsequence with rough sets," *Signal Processing: Image Communication*, vol. 30, pp. 178–189, 2015.
- [9] F. F. M. Ghaleb, E. A. Youness, M. Elmezain, and F. S. Dewdar, "Hand gesture spotting and recognition in stereo color image sequences based on generative models," *International Journal of Engineering Science and Innovative Technology*, vol. 3, no. 1, pp. 78–88, 2014.
- [10] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, "A unified framework for gesture recognition and spatiotemporal gesture segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1685–1699, 2009.
- [11] D. Frolova, H. Stern, and S. Berman, "Most probable longest common subsequence for recognition of gesture character input," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 871–880, 2013.
- [12] L. Liu, S. Yang, and D. Wang, "Particle swarm optimization with composite particles in dynamic environments," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 6, pp. 1634–1648, 2010.
- [13] J. Beh, D. Han, and H. Ko, "Rule-based trajectory segmentation for modeling hand motion trajectory," *Pattern Recognition*, vol. 47, no. 4, pp. 1586–1601, 2014.
- [14] H.-K. Lee and J. H. Kim, "An HMM-Based threshold model approach for gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 961–973, 1999.
- [15] A. D. Wilson and A. F. Bobick, "Parametric hidden Markov models for gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 884–900, 1999.
- [16] A. Just and S. Marcel, "A comparative study of two state-of-the-art sequence processing techniques for hand gesture recognition," *Computer Vision and Image Understanding*, vol. 113, no. 4, pp. 532–543, 2009.

- [17] M. Ángel Bautista, A. Hernández-Vela, V. Ponce et al., “Probability-based dynamic time warping for gesture recognition on RGB-D data,” in *Advances in Depth Image Analysis and Applications*, vol. 7854, pp. 126–135, Springer, 2013.
- [18] S. Zhou, F. Fei, G. Zhang et al., “2D human gesture tracking and recognition by the fusion of MEMS inertial and vision sensors,” *IEEE Sensors Journal*, vol. 14, no. 4, pp. 1160–1170, 2014.
- [19] B. Morris and M. Trivedi, “Learning trajectory patterns by clustering: experimental studies and comparative evaluation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 312–319, Miami, Fla, USA, June 2009.
- [20] J. Kennedy and R. C. Eberhart, “A discrete binary version of the particle swarm algorithm,” in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 5, pp. 4104–4108, Orlando, Fla, USA, October 1997.

Research Article

An Automatic Cognitive Graph-Based Segmentation for Detection of Blood Vessels in Retinal Images

Rasha Al Shehhi, Prashanth Reddy Marpu, and Wei Lee Woon

Masdar Institute of Science and Technology, Abu Dhabi, UAE

Correspondence should be addressed to Rasha Al Shehhi; ralshehhi@masdar.ac.ae

Received 28 February 2016; Revised 10 April 2016; Accepted 21 April 2016

Academic Editor: Daniel Zaldivar

Copyright © 2016 Rasha Al Shehhi 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 hierarchical graph-based segmentation for blood vessel detection in digital retinal images. This segmentation employs some of perceptual Gestalt principles: similarity, closure, continuity, and proximity to merge segments into coherent connected vessel-like patterns. The integration of Gestalt principles is based on object-based features (e.g., color and black top-hat (BTH) morphology and context) and graph-analysis algorithms (e.g., Dijkstra path). The segmentation framework consists of two main steps: preprocessing and multiscale graph-based segmentation. Preprocessing is to enhance lighting condition, due to low illumination contrast, and to construct necessary features to enhance vessel structure due to sensitivity of vessel patterns to multiscale/multiorientation structure. Graph-based segmentation is to decrease computational processing required for region of interest into most semantic objects. The segmentation was evaluated on three publicly available datasets. Experimental results show that preprocessing stage achieves better results compared to state-of-the-art enhancement methods. The performance of the proposed graph-based segmentation is found to be consistent and comparable to other existing methods, with improved capability of detecting small/thin vessels.

1. Introduction

Retinal vessel segmentation is a crucial step in analyzing fundus images of the eye for detection and diagnosis of many eye diseases. Some diseases such as glaucoma, diabetic retinopathy, and macular degeneration are very serious and might lead to blindness if they are not detected in time [1, 2]. The information about blood vessels, such as tortuosity and branching patterns, can not only provide information on pathological changes but also help to grade the disease severity and automatically diagnose the disease.

Although retinal vessel segmentation has been widely studied, it is still a challenging problem because of three main reasons. First, the quality of retinal images is highly variable and the segmentation methods face the challenge of low contrast or high homogeneity of illumination conditions [3–6]. Second, the complexity of vascular structures (different scales and orientations) means that most existing methods find it difficult to enhance multiscale vessel-like structures with various linear orientations [5–12]. Third, finding the most

optimum model or method which is the most appropriate for variety of data is very difficult [13–15].

The studies of retinal images can be classified into pattern recognition (machine learning/model-based) [6, 16, 17], mathematical morphology [6, 18], kernel-based analysis [10, 11, 17] and tracking-based/path-based (Artificial Intelligent, AI) methods [5, 19, 20]. Here, morphological and AI methods are further discussed because they are more related to presented work in this paper.

Morphological methods examine the geometric vessel-like structure of retinal image by probing it with small patterns called structuring elements (SE) of predefined size and shape. Due to sensitivity of vessel-like patterns to different scales and orientations, most methods use multiscale or/and multiorientation structuring elements [18, 21, 22], such as multistructure morphological operators [8, 12], and multiscale white top-hat with linear structuring elements [9]. One of the challenges is that there are several structures in retinal images such as optical disk, exudates, microaneurysms, and hemorrhages, which degrade the performance

of vessel detection methods. To overcome this problem, a number of approaches have been proposed in order to decompose components in retinal images. In [6], Morphological Component Analysis (MCA) has been proposed to separate some components such as lesions from vessels.

Tracking-based/path-based methods use regional information (a single vessel rather than the entire vasculature) to find the shortest/cheapest path that matches a vessel profile. The main advantage of this approach is that it provides precise vessel width, unlike other methods. Nowadays, there are many studies which follow this approach, for example, Dijkstra shortest path for vessel patterns [19], graph-cut [5], Bayesian-based tracking [20], and graph-analysis [23].

In this work, we propose a perceptual graph-based segmentation method. The complete framework consists of two stages. The first stage (preprocessing) removes noise as well as unwanted regions such as optical disk and surrounded darker background and produces a higher contrast vessel image. The second stage (segmentation) converts an image into connected graphical layer where each pixel is presented as a node and its spatial/spectral properties are used to merge pixels (nodes) to construct more semantic objects in higher-connected graphical layers. Gestalt perceptual principles, that is, similarity, closure, continuity, and proximity of spatial/spectral properties of nodes, are employed to assemble smaller parts that are most likely to represent a coherent connected vessel-like pattern.

The experimental evaluation is carried out to test the behavior of segmentation algorithm using the standard datasets such as DRIVE, ARIA, and STARE (details of these datasets are given in Section 4.1) using the following major criteria: sensitivity (Se), specificity (Sp), accuracy (Acc), and area under curve (AUC).

In the following, the idea of perceptual Gestalt principles in image segmentation is introduced in Section 2. Section 3 illustrates the proposed hierarchical graph-based segmentation framework in detail. First, it presents the preprocessing part, which includes filtering-based inhomogeneity correction using Gaussian filter, followed by morphology-based illumination enhancement method. Second, it shows the segmentation part, which is based on integrating the idea of perceptual Gestalt principles into object-based features to merge segments. Then, Section 4 presents the datasets, experimental metrics, and results of segmentation. Finally, the conclusions are drawn and some ideas for future work are presented in Section 5.

2. Contribution of Current Work

The main contribution of this work is to introduce perceptual Gestalt (form, grouping) principles [24–28] of some middle-level image features into graph-based segmentation to discriminate the connected coherent vessel-like patterns from background.

Four Gestalt principles are proposed, inspired from similarity, closure, continuity, and proximity. The theory behind Gestalt grouping is based on how human vision performs perceptual grouping to assemble parts of an image that most

likely represent a single object in the scene. Similarity can be used to group segments into one object depending on the number of similar factors such as color, size, and shape. Proximity rule is employed by assembling different parts which are close to each other to present one object. Good continuity is the tendency of elements to be grouped to form smooth contours depending on the number of factors such as orientation of local elements, contour length, and curvature properties. The principle of closure refers to tendency to see an element/object as a complete form or figure, ignoring gaps and incomplete contour lines. It is not necessary to create triangles, circle, and so forth, but it fills the missing information to create familiar shapes [25, 26, 28].

The integration of perceptual Gestalt principles into graph-based segmentation, from computational view, is an important stage to reduce required visual processes to interpret an input image by converting a fully connected layer into locally connected layer [29]. Moreover, such integration helps to cope with undersegmentation/oversegmentation in image layer(s) or between layers.

In this work, the perceptual principles are employed as follows. The first level, defined as color-layer, is built by grouping pixels based on similarity of color between each pixel and its 8-connected neighborhood (Gestalt similarity in illumination characteristics). The second layer, called as black top-hat (BTH) layer, is constructed by grouping adjacent objects, which most likely represent vessel-like shape after applying BTH morphological operator (Gestalt closure in connected components of most common label in BTH-layer). The final Dijkstra-layer is created by conjunction of adjacent objects which have a high probability of constructing connected objects (Gestalt continuity in Dijkstra tracking path). Gestalt proximity is employed by considering 8-connected neighborhood as a Gestalt connectivity patch in all layers.

3. The Proposed Method

The proposed framework (Figure 1) comprises two major stages: preprocessing and segmentation. Preprocessing (green rectangles: A–C) consists of filtering-based inhomogeneity correction and morphology-based illumination enhancement (Section 3.1). Hierarchical graph-based segmentation (red rectangles: 1D–3D) is based on construction of five layers based on the number of objects in retinal images: original RGB image (largest), ROI (region of interest), color, followed by black top-hat (BTH), and finally Dijkstra-layers (smallest) (Section 3.2).

3.1. Preprocessing. Preprocessing involves two main steps to produce more effective feature image which shows a high contrast between vessel and nonvessel objects to facilitate the segmentation. The first step is to remove the effect of background variations by nonuniform illumination and the second step is to eliminate the complexity of vascular structure because of multiple scales and orientations.

In this work, we choose the green channel image for preprocessing, as it exhibits the best vessel/no-vessel contrast in retinal images. The red channel can be saturated and blue

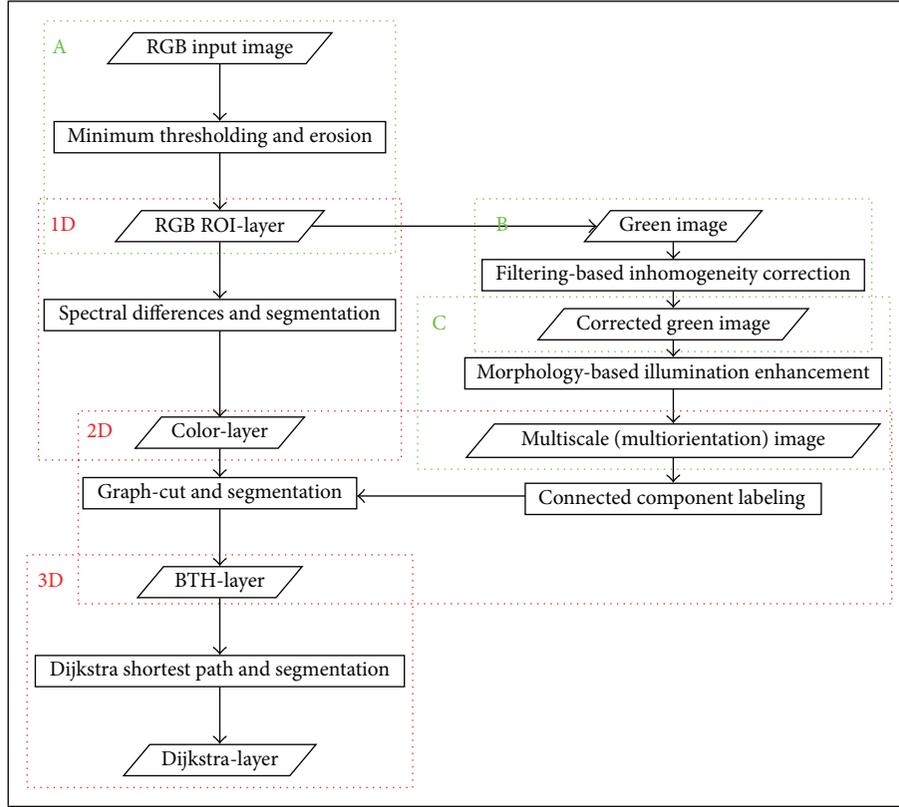


FIGURE 1: Framework of perceptual hierarchical graph-based segmentation. Preprocessing stage is presented in green rectangles. Multiscale graph-based segmentation is shown in red rectangles.

channel has poor dynamic ranges [16]. In addition, using only the green channel decreases the computational time compared to processing all RGB channels. We first extract the region of interest (ROI) representing the fundus, which is a circle-like shape/nonblack region at the center of retinal images without including the surrounding black background. We eliminate the influence of the background by masking it to accelerate further processing stages by focusing only on pixels/objects in ROIs. To find ROI, simple minimum threshold for all red, green, and blue channels is applied to remove unwanted background from RGB channel. However, some missed labeled pixels are created on retinal foreground and background. These noises are eliminated by morphological erosion [30–32], where ROI is shrunk into center of retinal images (Figure 1(A)).

3.1.1. Filtering-Based Inhomogeneity Correction (Figure 1(B)). Due to inhomogeneous light conditions, retinal images may contain background (nonvessels) with high similarity to foreground (vessels), which will degrade the performance of the segmentation method. Therefore, it is important to remove effects of the varying illumination conditions. Zhao et al. [5] applied Retinex theory, adapted from the field of computer vision, to remove unwanted illumination effects by component-wise multiplication of reflected green with green illumination channel ($R_G \times G$). R_G is component-wise log-subtraction bilateral blurring of green channel $I(B_G)$ from

green channel (G). Imani et al. [6] used reflectance component of Retinex theory to reduce illumination differences, but with component-wise subtraction of median blurring of green channel. Some other studies used Contrast-Limited Adaptive Histogram Equalization (CLAHE) method instead of global enhancement methods such as histogram equalization and gamma correction; however, its local enhancement is uniform regardless of whether it is foreground or background [3, 4].

In this work, we use the component-wise subtraction of background from the selected channel to remove the light variations. Let x be a pixel of a green channel. The retinal image background (f_B) is created by applying low-pass Gaussian blurring of green channel (f_G) with a reasonable filter size (e.g., $w = 61 \times 61$) to eliminate the effect of the brightest region from retinal images (optical disk). The corrected image (f_C) is computed by subtracting f_B from f_G . This is given as

$$f_C(x) = f_G(x) - f_B(x). \quad (1)$$

Figure 2 illustrates a comparison between previously mentioned correction methods using selected examples from DRIVE, ARIA, and STARE datasets. In general, all methods enhance image contrast, but there are still large areas of homogeneity. Component-wise subtraction of Gaussian blurring is not the best method; however, it has succeeded in eliminating the most noisy part (optical disk) and enhances

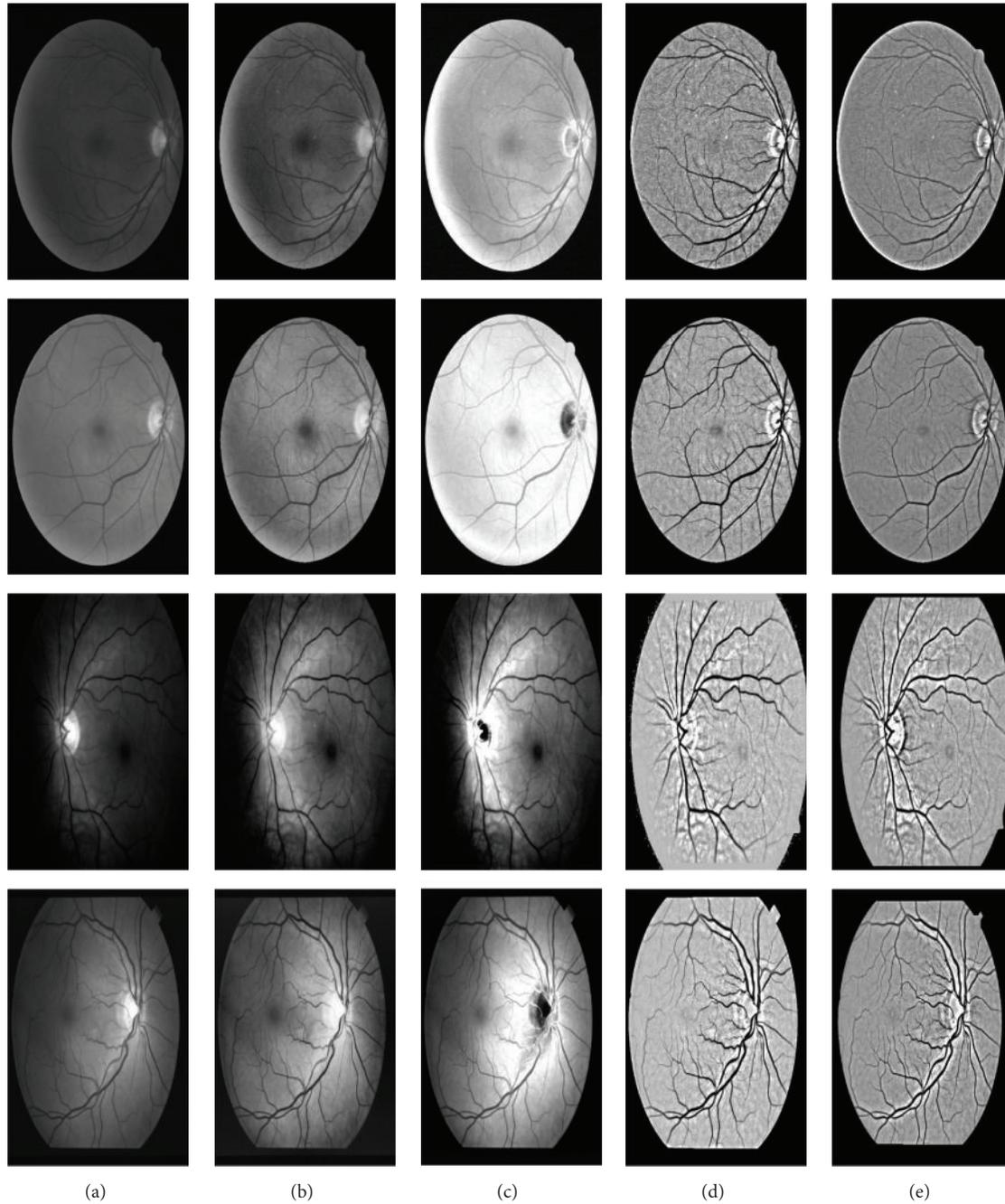


FIGURE 2: Inhomogeneity correction results on selected images. (a) shows green channel of randomly chosen images from the DRIVE training, DRIVE testing, ARIA, and STARE datasets, respectively. (b) shows CLAHE results [3, 4]. (c) shows results of Retinex theory on green channel [5]. (d) presents difference between green channel and median blurring of green channel [6]. (e) presents results of our correction method.

the contrast between vessels and background. This can be especially noticed in ARIA and STARE images after applying morphology enhancement, as depicted in Figure 4. As a consequence, the vessels can be easily identified from the background. Component-wise correction of bilateral filtering has comparable results after applying morphology enhancement, because it is an edge-preserving smoothing filter that maintains edges of vessels [33].

3.1.2. Morphology-Based Illumination Enhancement (Figure 1(C)). Mathematical morphology is a nonlinear method which uses the concepts of set, topology, and geometry to analyze geometrical structures (e.g., shape and form of objects) in images. It examines the geometric structure of an image by probing with structuring elements (B) [30–32].

In this work, black top-hat (BTH) morphological operator is used. This is because the BTH operator is the most

applicable method to extract image structure under lower illumination conditions [30], which is the case in blood vessel detection application. The BTH is obtained by subtracting morphological closing image $\phi(f(x))$ of corrected image from corrected image $f(x)$, as in (2). The morphological closing $\phi(f(x))$, which is dilation $\delta_B(f)$ followed by erosion $\varepsilon_B(f)$, acts as a shape filter and preserves objects having relevant structure in image (3). This is given as

$$\text{BTH}(f) = \phi(f(x)) - f(x), \quad (2)$$

$$\phi(f) = \varepsilon_B(f)(\delta_B(f)). \quad (3)$$

All vessel-like patterns are components of linear-shape structures with various horizontal, vertical, and diagonal orientations. To address these differences, we suggest the following approaches.

(i) *Multiscale Multiorientation BTH.* The BTH is adaptively computed by probing the corrected image with multiple linear structuring elements which have a variety of angular orientations. Therefore, a set of linear structuring elements, where each is a matrix representing a line with 3, 5, 9, 15, and 21 pixels of length and rotated in steps of $\pi/8$, is used for morphological BTH. As a result, each isolated BTH along each direction will brighten the vessels in that direction, providing that the length of B is large enough to extract the vessel with largest diameter. Finally, we consider the average of the first three maximum BTH results because the first three BTHs present the highest differences between vessel/nonvessel patterns, as the following:

$$\text{BTH} = \frac{\sum_1^3 \max \text{BTH}_s^\theta(f)}{3}, \quad (4)$$

where s and θ are scale and orientation of linear structuring elements B .

(ii) *Multiscale BTH.* Multiscale BTH is defined as an average of morphological BTHs obtained from probing corrected image with size of 3, 5, 7, 9, and 11 of elliptical-structuring elements B , unlike the previous method which uses linear structuring elements in different directions, given by

$$\text{BTH} = \frac{\sum_1^5 \text{BTH}_s(f)}{5}. \quad (5)$$

In this paper, two other enhancement methods are selected for comparison: phase-based method [5] and wavelet-based method [10]. In order to reproduce results, the parameters used for these filters are the same as recommended in the corresponding literature [5, 10]. Figure 3 shows the results of applying multiscale BTH, multiscale multiorientation BTH, phase-based method, and wavelet-based method. It can be seen clearly that illumination contrast using wavelet-based method is poor compared to phase-based method and the proposed method, where the vessel-like structure is more distinguishable. However, multiscale BTH method produces more consistent results at optical disk and foveal areas compared to other parts of the vessels. Figure 4 shows the result

of applying multiscale BTH and multiscale multiorientation BTH to different correction methods, that is, CLASHA [3, 4], Retinex [5], difference of green and its median blurring [6], and proposed correction method. In summary, it seems that applying morphological operators over all correction images successfully enhances the contrast of vessels. The results of multiscale BTH over proposed correction method seem to be efficient in removing inhomogeneity within the image including optical disk and fovea regions.

3.2. *Hierarchical Graph-Based Segmentation (Figure 1(D)).* In this paper, a cognitive vision approach to graph-based image segmentation is proposed by employing perceptual knowledge of contextual features to provide semantically vessel-like patterns. Moreover, it decreases the required computational cost by processing perceptual features instead of processing the fully connected image in blood vessel applications.

Graph-based segmentation represents the input image as a weighted graph $G = \{V, E, W\}$, where all vertices/nodes $v_i \in V$ correspond to a pixel/region of the image $V = \{v_1, v_2, \dots\}$ and edges $e(v_i, v_j) \in E$ correspond to pair of neighboring vertices/nodes. Each edge $e(v_i, v_j) \in E$ has a corresponding weight $w(v_i, v_j)$ which represents the dissimilarity between adjacent vertices/nodes i and j . The dissimilarity between edges is often defined based on selected properties relevant to the application (e.g., similarity of color and shape) [34–36].

In this work, we build an undirected graph with multilevel layers, where the number of vertices/nodes is hierarchically reduced from top-layer to the next bottom-layer because nodes are merged to construct objects with more semantic interpretation from visual perception view. There are two most important questions in graph-based algorithms to be considered in image segmentation. First is the weighting function that presents spectral or/and spatial relationship between adjacent nodes. Second is the merging/homogeneity criteria to group adjacent vertices/nodes to have connected components. The graph starts with color-layer and BTH-layer, followed by Dijkstra-layer. The design of each layer is illustrated in the following sections.

3.2.1. *Gestalt Similarity and Proximity in Contextual Color Features (Figure 1(ID)).* In order to build spectral level, the graph-based framework proposed in [35] is applied. It translates Gaussian smoothed input image into a graph, where each pixel $v_i \in V$ is mapped to vertice/node and each edge $e_k = (v_i, v_j) \in E$ reflects spectral relationships between adjacent pixels. We consider 8-connected neighborhood as a Gestalt connectivity patch. The initial weighting function is Euclidean distance between red, green, and blue components between two adjacent vertices/nodes v_i and v_j . This is given as

$$\begin{aligned} w(v_i, v_j) &= \sqrt{(R_{v_i} - R_{v_j})^2 + (G_{v_i} - G_{v_j})^2 + (B_{v_i} - B_{v_j})^2}. \end{aligned} \quad (6)$$

The spectral vertices/nodes are hierarchically merged based on degree of spectral similarity (for more details of the

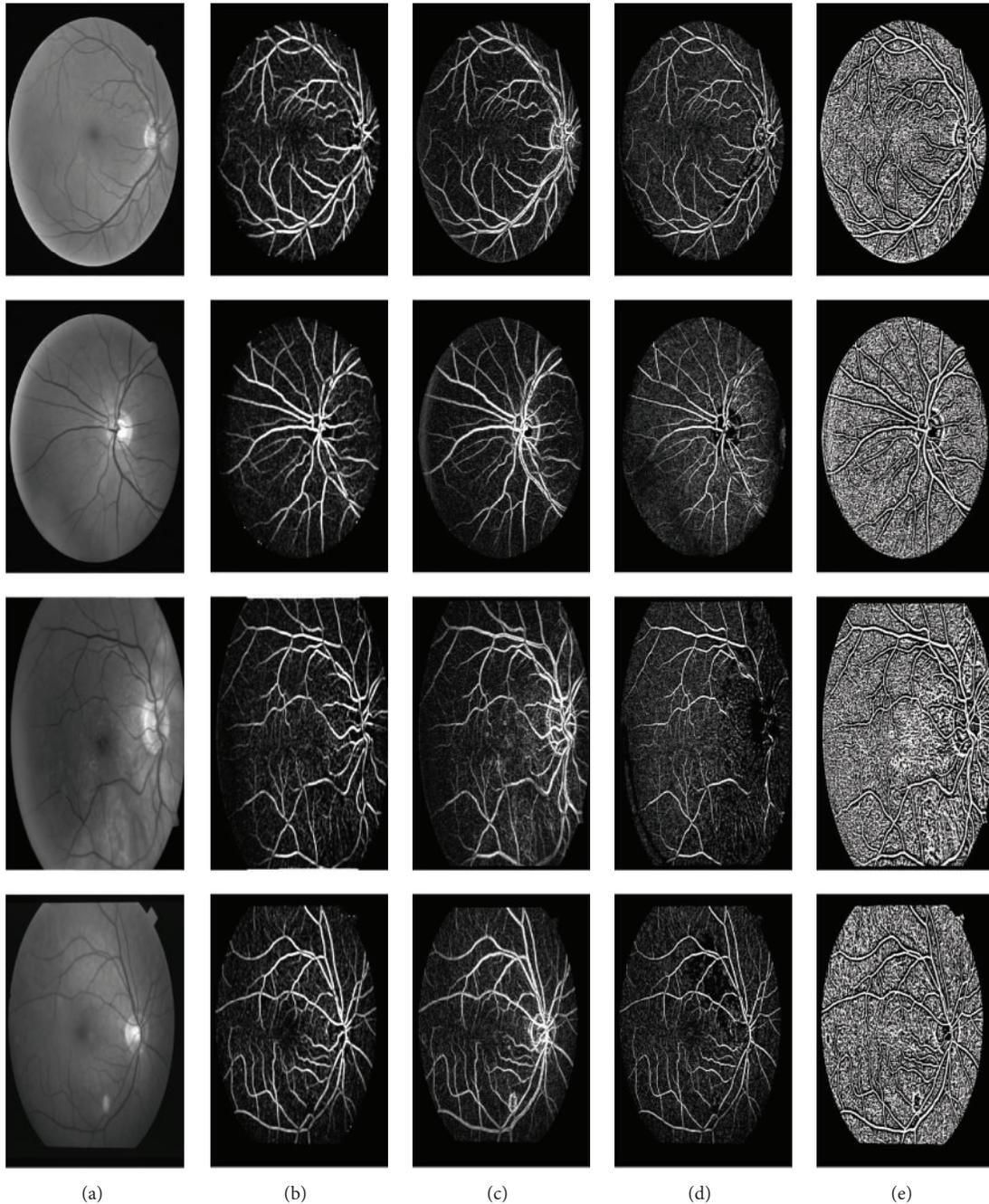


FIGURE 3: Illustrative comparison of enhancement effects in selected images from the DRIVE training, DRIVE testing, ARIA, and STARE images. The green channels of selected images are presented in (a). (b) shows the results of applying multiscale BTH. Results of multiscale multiorientation BTH are presented in (c). (d) shows local-phased enhancement after applying Retinex to green channels [5]. (e) shows results of applying wavelet-based enhancement results [10].

algorithm, read from [35]). The output of spectral-grouping is not robust enough to perceptually interpret resulting segmented regions as semantic components, as depicted in Figure 5(c). High-level criteria may efficiently enhance results by grouping/splitting components into more meaningful spatial structures starting from spectral components instead of pixels. Therefore, grouping in higher layers is similar to the spectral-layer, but with other weighting and merging criteria for different measures of similarity. At each stage,

components are merged iteratively from small to large until convergence (no more merging). The stop condition is based on prior knowledge and it is used to prevent oversegmentation/undersegmentation.

3.2.2. Gestalt Closure and Proximity Based on Contextual BTH Features (Figure 1(2D)). A sequential connected component labeling approach [37, 38] is employed to separate

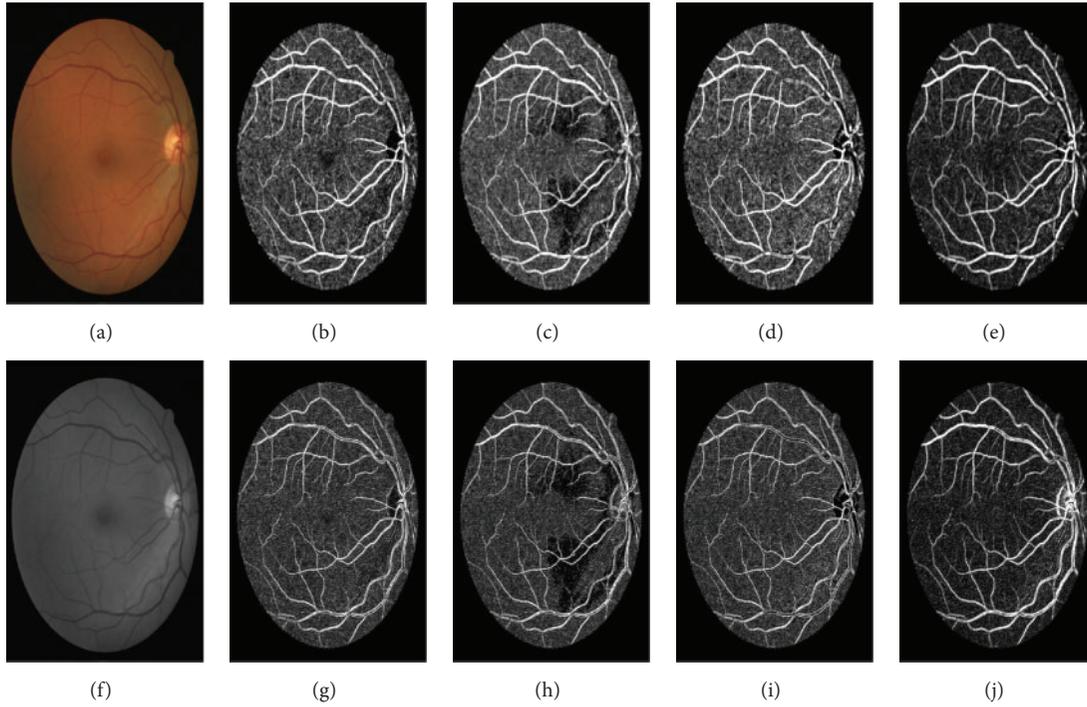


FIGURE 4: Enhancement results produced by applying multiscale BTH ((a)–(e)) and BTH of multiscale multiorientation structuring elements ((f)–(j)) to CLAHE channel ((b) and (g)) of green channel [3, 4], Retinex channel ((c) and (h)) [5], median channel ((d) and (i)) [6], and proposed method ((e) and (j)).

vessel-like components from background components by scanning all pixels in vessel-layer and consequently labeling each connected component. For convenience, we consider the case of 8-connected neighborhood (Gestalt proximity patch). The suggested labeling threshold is based on ratio of mean average over standard deviation average of vessel-layer with 10 homogeneous pixels as the minimum number of pixels in one component.

We consider the labeled-component with highest number of pixels as background (label 0) and other components as foreground components (labels 1, 2, 3, 4, 5, etc.). On the other hand, nonmasked area is typically left unchanged by the labeling process. As a result, background label (L_B) is used to cut edges of spectral components or vertices from spectral components (see (7)) to build BTH-layer where the number of nodes in this layer is lower than the number of nodes in spectral-layer, as depicted in Figure 5(d):

$$\text{BTH}(v) = \begin{cases} \text{vessel}, & v \in L_F, \\ \text{no-vessel}, & v \in L_B. \end{cases} \quad (7)$$

3.2.3. Gestalt Continuity and Proximity Based on Contextual Features (Figure 1(3D)). The BTH-layer is updated by assigning a new weight for edges between BTH-spectral connected components. The weight is based on the smallest Euclidean distance between vertices of all adjacent components.

The development of our continuity feature is motivated by the need for a representation of long contour representation

suitable to visually perceive vessel-like patterns as continuous irregular lines. In order to apply continuity principle, Dijkstra algorithm [19, 39] is employed within window size $w = 50 \times 50$ by considering each first component in BTH-layer block as a source point of the graph-path and each furthest component in window $w = 50 \times 50$ as target point of graph-path. All vertices in Dijkstra path from source to target are iteratively merged to be one component, until convergence of vessel and nonvessel regions (Figure 5(e)).

Figure 5 shows some examples from standard datasets. The spectral segmentation is not sufficient to obtain meaningful structures, because lighting contrast of vessel-like/non-vessel-like patterns is low. Therefore, the results are with undersegmented/oversegmented regions, so higher-level criteria should be used efficiently to group or split components into more meaningful spatial structures (BTH). This shows the importance of using shape features to eliminate wrongly labeled vessel-like patterns. On the other hand, other properties such as asymmetry and length/width [40] help to decrease the number of nodes in BTH-layer; however, they are not enough to find connectedness of nonadjacent nodes in BTH-layer.

As a result, Graph-based characteristics (Dijkstra algorithm) are introduced in image segmentation to determine connectedness of nonneighbor nodes, which can be integrated to build a complete contour. Table 1 presents the number of components in vessel multilayer graph, starting from initial green channel, ROI image, and BTH and Dijkstra-layers.

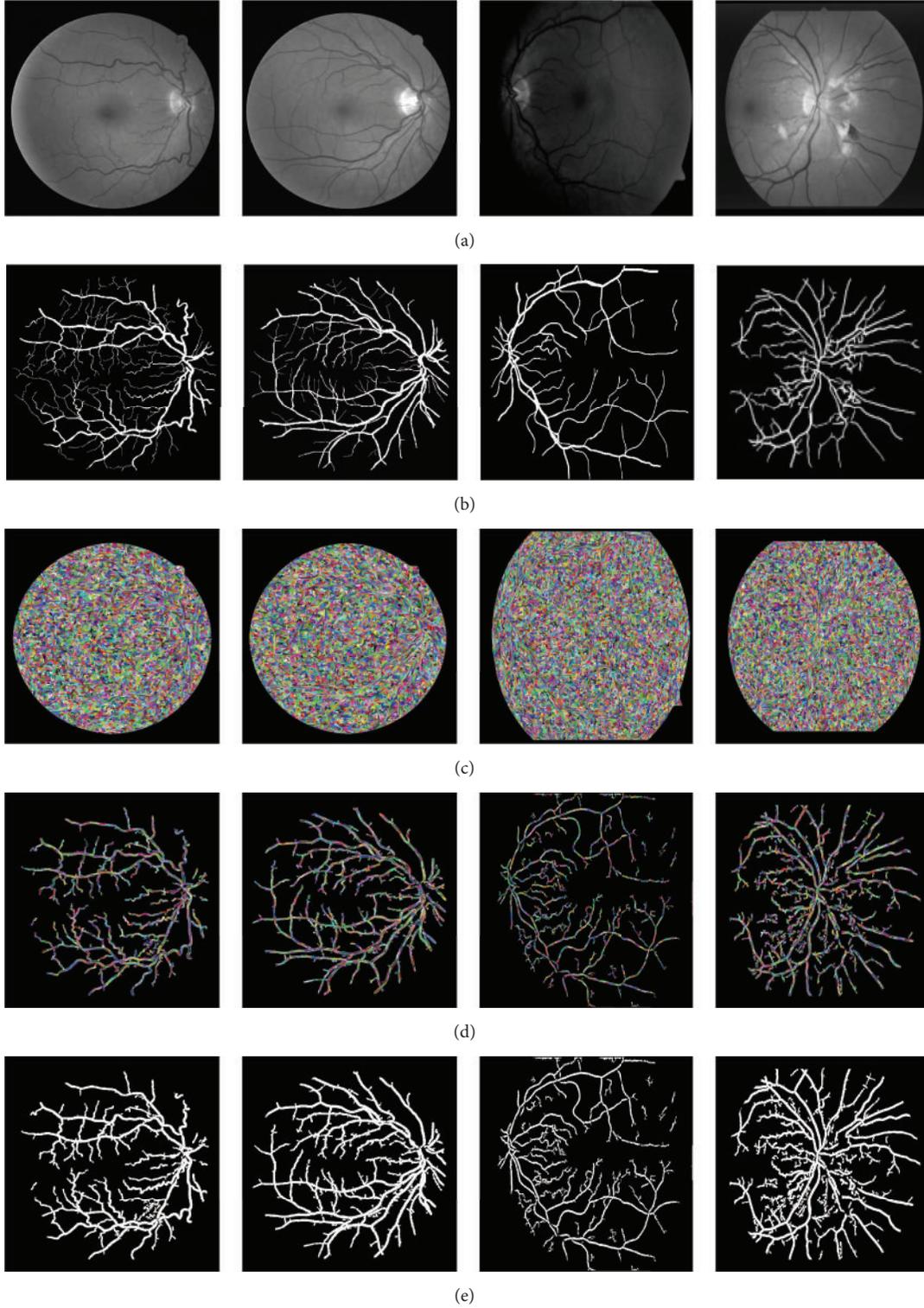


FIGURE 5: Demonstrative results of each stage of applying graph-based segmentation. (a) shows some selected images from the DRIVE training, DRIVE testing, ARIA, and STARE datasets. Gold standards are shown in (b). (c) presents spectral-layer. The BTH-layer after applying graph-cut to vessel-layer (multiscale BTH) is illustrated in (d). (e) shows the final-layer after applying Dijkstra path.

TABLE 1: Illustrative study of the number of components from fully connected to locally connected layer of selected examples from the DRIVE training, DRIVE testing, ARIA, and STARE datasets.

| Datasets | DRIVE training | DRIVE testing | ARIA | STARE |
|------------------------|----------------|---------------|--------|--------|
| Initial layer | 329960 | 329960 | 442368 | 423500 |
| Initial layer with ROI | 210702 | 208174 | 350013 | 288761 |
| Spectral-layer | 9874 | 9816 | 15094 | 12917 |
| BTH-layer | 3285 | 3719 | 3130 | 4968 |
| Dijkstra-layer | 2 | 2 | 2 | 2 |

4. Experimental Evaluation

We have employed three public retinal image datasets to evaluate the proposed segmentation framework. In this section, a brief introduction to these datasets is provided in Section 4.1; evaluation criteria including preprocessing and segmentation criteria are defined in Section 4.2, followed by experimental results in Section 4.3.

4.1. Data. We obtained human retinal images from publicly available datasets: DRIVE, ARIA, and STARE. All datasets consist of RGB components of retinal images with their corresponding ground truth images where blood vessel-like structures are segmented. These datasets are selected because of availability of gold standard from manual annotations of retinal vessels by experts.

DRIVE (Digital Retinal Images for Vessel Extraction). It consists of training and testing image with 565×584 pixels, which are obtained from a diabetic retinopathy screening program in the Netherlands. The set of 40 photographic images have been randomly selected, 33 do not show any sign of diabetic retinopathy, and 7 show signs of mild early diabetic retinopathy. The manual segmentation of set A is used as ground truth. The DRIVE dataset is available at <http://www.isi.uu.nl/Research/Databases/DRIVE/>.

ARIA (Automated Retinal Image Analysis). It consists of three groups: 92 images of age-related muscular degeneration, 59 images of patients with diabetes, and 61 images of healthy eyes, collected by St. Paul's Eye Unit and University of Liverpool. Each image was captured at a resolution of 768×576 pixels. The manual segmentation from observer DGP is used as ground truth. The ARIA dataset is available at <http://aria.cvs.rochester.edu/#&panel1-1>.

STARE (Structural Analysis for the Retina). It consists of 20 images with 10 images showing evidence of pathology, obtained from University of California. Each image was captured by 8 bits per plane at 605×700 pixels. Hand-labeled vessel networks, provided by Adam Hoover (1st manual) and Valentina Kouznetsova (2nd manual), are used as ground truth. The STARE datasets are available at <http://www.ces.clemson.edu/~ahoover/stare/probing/index.html>.

4.2. Evaluation Measures

4.2.1. Preprocessing Measures. In order to evaluate contrast enhancement, several objective methods are proposed: Contrast Improvement Index (CII), Peak Signal-to-Noise Ratio (PSNR), and Mean Structural Similarity (M_{SSIM}) [8, 45, 46].

The CII is ratio of contrast of the enhanced and original images. CII is computed as follows:

$$CII = \frac{C_{enh}}{C_{original}}, \quad (8)$$

$$C = \frac{fg - bg}{fg + bg},$$

where C presents the contrast between foreground (vessel) and background (retinal regions except vessels). fg and bg are mean gray-level value of foreground and background, respectively. The larger C and consequently the larger CII, the more obvious the difference between foreground and background (higher contrast and better enhancement). The PSNR measures intensity changes of original and enhanced images based on Mean Square Error (MSE), as follows:

$$MSE = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \|I_{original}(i, j) - I_{enh}(i, j)\|^2, \quad (9)$$

$$PSNR = 10 * \log_{10} \left(\frac{\max_I^2}{MSE} \right).$$

To compute M_{SSIM} , the image is decomposed into k blocks of size $n \times m$ and, for each block, SSIM is computed as follows:

$$SSIM = \frac{(2\bar{x}\bar{y} + C_1)(2\sigma_{xy} + C_2)}{(\sigma_x^2 + \sigma_y^2 + C_2)(\bar{x}^2 + \bar{y}^2 + C_1)}, \quad (10)$$

where \bar{x} , \bar{y} , σ_x , and σ_y are means and standard deviations in original and enhanced blocks. σ_{xy} corresponds to covariance measure, $C_1 = (0.01 \times 255)^2$, and $C_2 = (0.03 \times 255)^2$.

4.2.2. Segmentation Measures. Four common metrics are employed to measure the performance of the proposed segmentation: sensitivity (Se) or True Positive Rate (TPR), specificity (Sp) or False Positive Rate (FPR), accuracy (Acc), and area under curve (AUC). Sensitivity indicates the proportion of vessel pixels identified by the proposed method that coincide with vessel pixels in ground truth images (11), while specificity is the proportion of detected pixels by the proposed method that are not vessels in ground truth images (12). Accuracy indicates the proportion of vessel/nonvessel patterns that are correctly identified in terms of the total number of pixels in retinal images (13). ROC is obtained by plotting the fraction of Se/TPR versus Sp/FPR (14). The closer

TABLE 2: Graph-based segmentation performance without and with correction methods on DRIVE, ARIA, and STARE datasets, respectively. 2nd row: CLAHE ($w = 3 \times 3$) [3, 4, 41]. 3rd row: Retinex ($w = 3 \times 3$, spatial spread based on low-pass filter $\sigma_d = 0.3$, and geometric spread of the image intensity $\sigma_r = 0.3$) [5]. 4th row: subtraction median blurring image of green ($w = 31 \times 31$) from green image [6]. 5th row: subtraction low-pass Gaussian blurred image of green ($w = 61 \times 61$) from green image. Se: sensitivity, Sp: specificity, Acc: accuracy, and AUC: area under curve.

| Datasets | Correction methods | Evaluation metrics | | | |
|----------|--------------------|--------------------|-------|--------------|---------|
| | | Se | Sp | Acc | AUC |
| DRIVE | Without | 0.896 | 0.779 | 0.789 | 0.837 |
| | CLAHE | 0.841 | 0.867 | 0.865 | 0.806 |
| | Retinex | 0.804 | 0.934 | 0.923 | 0.854 |
| | Green-MF (Green) | 0.911 | 0.619 | 0.645 | 0.764 |
| | Green-GF (Green) | 0.921 | 0.702 | 0.931 | 0.0.869 |
| ARIA | Without | 0.743 | 0.913 | 0.898 | 0.828 |
| | CLAHE | 0.781 | 0.823 | 0.819 | 0.802 |
| | Retinex | 0.768 | 0.701 | 0.707 | 0.734 |
| | Green-MF (Green) | 0.794 | 0.667 | 0.679 | 0.731 |
| | Green-GF (Green) | 0.721 | 0.944 | 0.924 | 0.832 |
| STARE | Without | 0.629 | 0.924 | 0.850 | 0.774 |
| | CLAHE | 0.643 | 0.902 | 0.838 | 0.770 |
| | Retinex | 0.650 | 0.894 | 0.839 | 0.769 |
| | Green-MF (Green) | 0.670 | 0.866 | 0.821 | 0.769 |
| | Green-GF (Green) | 0.563 | 0.975 | 0.879 | 0.774 |

the curve approaches the top left corner with AUC close to 1, the better the performance of the proposed method is [47]:

$$Se = \frac{TP}{TP + FN}, \quad (11)$$

$$Sp = \frac{TN}{TN + FP}, \quad (12)$$

$$Acc = \frac{TP + TN}{TP + FP + TN + FN}, \quad (13)$$

$$AUC = \frac{Se + Sp}{2}, \quad (14)$$

where TP is defined as the number of vessel pixels correctly detected in the retinal images, FP is defined as the number of nonvessel pixels detected as vessels, TN is the number of nonvessel pixels correctly detected, and FN is the number of vessel pixels detected as nonvessel pixels.

4.3. Results. In order to evaluate the efficiency of the proposed segmentation method, we use two approaches. First, each individual stage of the proposed method is evaluated by its comparable stages of other methods considered in this study (i.e., inhomogeneity correction and illumination enhancement) across all three datasets. Second, the performance of the proposed segmentation is compared with other works across DRIVE and STARE datasets.

4.3.1. Inhomogeneity Correction Assessment. The optical disk and fovea area contribute to most of the false detection of vessel pixels in most blood vessel detection frameworks [9,

10, 48]. Subtracting green channel from low-pass Gaussian blurring of green channel increases the contrast between optical disk/foveal area and vessel pixels. Therefore, the noisy area will not be enhanced after morphology enhancement, especially after multiscale BTH. Figure 6 presents an example of applying different correction methods, mentioned in Section 3.1.1, to one of the images from DRIVE testing dataset. Our correction method (Figure 6(e)) shows a superior performance for the vessel subtraction and is more similar to ground truth image, compared to other methods. Table 2 shows that the accuracy of our correction method is the highest in all three datasets.

4.3.2. Illumination Enhancement Assessment. Figure 7 shows an example of applying enhancement methods to an image selected randomly from DRIVE testing dataset. The results of multiscale BTH outperform other enhancement methods. The false detection of segmentation method after applying multiscale multiorientation BTH is because it fails to remove optical disk and some exudates. Optic disk area, which can be seen as brightest, round, vertically slightly oval disk, can be easily preserved by multiscale multiorientation BTH, which identifies line-shape structures in vertical, horizontal, or diagonal angularity with varying sizes. Moreover, exudates can be identified as areas with varying size and shapes, where it is difficult to be eliminated by multiscale multiorientation BTH. In contrast, multiscale BTH maintains just all ellipse-shape structures regardless of their orientations and then finds magnitude of directional blurring of BTH. As depicted in Table 3, the accuracy of multiscale BTH is above 90% in all

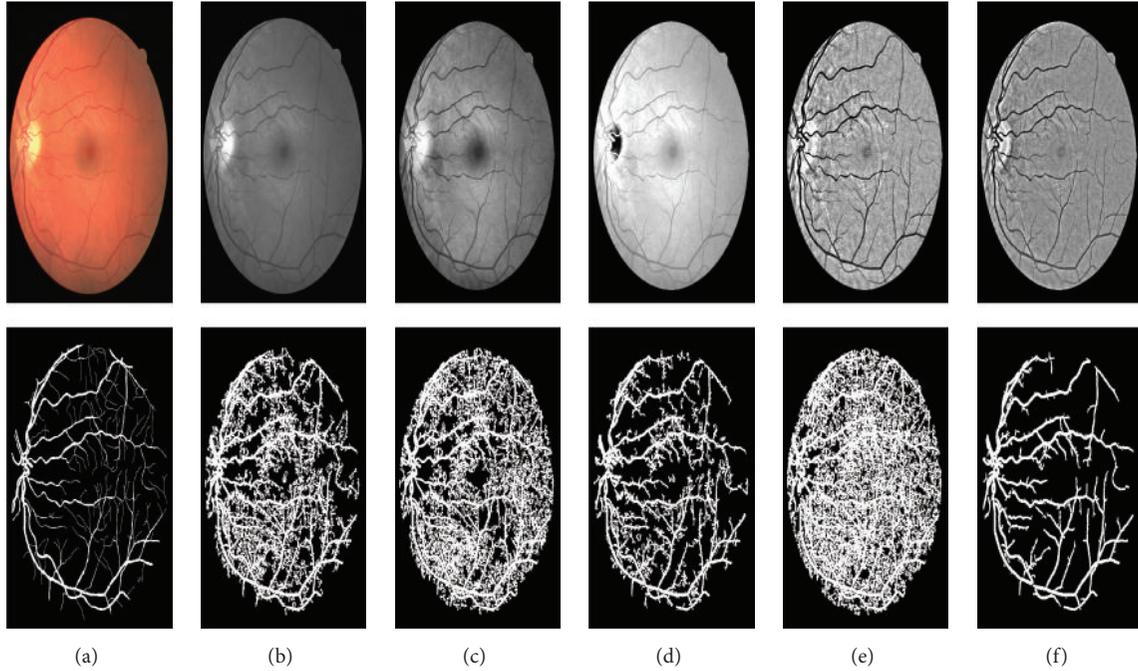


FIGURE 6: Demonstrative comparison of correction methods on one of the DRIVE testing images. The first row shows the results of inhomogeneity correction method and its consequence final segmentation in the second row. (a) RGB image and 1st manual. (b) Green channel without correction. (c) CLAHE correction. (d) Retinex correction. (e) Difference between green and median blurring of green channels. (f) Difference between green and Gaussian blurring of green channel.

TABLE 3: Graph-based segmentation performance of four enhancement methods: local phase-based [5], wavelet-based [10], multiscale multiorientation BTH, and multiscale BTH methods on DRIVE, ARIA, and STARE datasets, respectively. Se: sensitivity, Sp: specificity, Acc: accuracy, and AUC: area under curve.

| Datasets | Enhancement methods | Evaluation metrics | | | |
|----------|---------------------------------|--------------------|-------|--------------|-------|
| | | Se | Sp | Acc | AUC |
| DRIVE | Phase-based | 0.820 | 0.827 | 0.826 | 0.823 |
| | Wavelet-based | 0.827 | 0.768 | 0.773 | 0.798 |
| | Multiscale multiorientation BTH | 0.635 | 0.924 | 0.945 | 0.803 |
| | Multiscale BTH | 0.863 | 0.930 | 0.924 | 0.893 |
| ARIA | Phase-based | 0.469 | 0.818 | 0.785 | 0.644 |
| | Wavelet-based | 0.725 | 0.683 | 0.687 | 0.704 |
| | Multiscale multiorientation BTH | 0.813 | 0.893 | 0.871 | 0.828 |
| | Multiscale BTH | 0.746 | 0.947 | 0.920 | 0.841 |
| STARE | Phase-based | 0.432 | 0.776 | 0.855 | 0.699 |
| | Wavelet-based | 0.469 | 0.773 | 0.711 | 0.620 |
| | Multiscale multiorientation BTH | 0.660 | 0.944 | 0.888 | 0.803 |
| | Multiscale BTH | 0.606 | 0.977 | 0.910 | 0.800 |

datasets and consequently their AUC values are most closer to 1.

The suggested preprocessing enhancement method is tested by combining it with correction methods and comparing it with other enhancement methods, that is, local-phased and wavelet-based methods, in terms of the following criteria: CII, PSNR, and M_{SSIM} (Table 4). The CII and PSNR of multiscale BTH with low-pass Gaussian blurring correction are relatively largest. On the other hand, M_{SSIM} of both median and Gaussian corrections are comparable. The reason

is that both filtering methods succeed in efficiently removing noisy areas by filtering operators of large size.

4.3.3. Comparison between Proposed Segmentation and State-of-the-Art Methods. The segmentation performance of the proposed method in terms of sensitivity, specificity, accuracy, and area under curve is compared with other state-of-the-art methods (matched filtering, supervised, unsupervised, and artificial methods) in the most public datasets: DRIVE

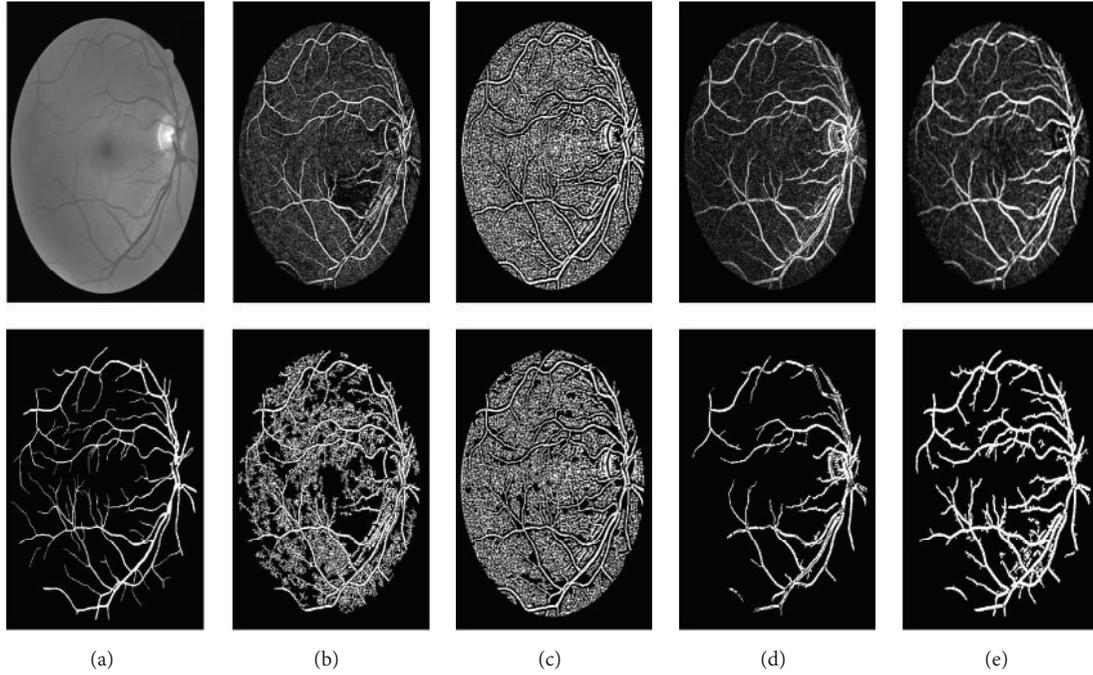


FIGURE 7: Comparison between illumination enhancement methods and their consequent graph-based segmentation results on one of the selected images from DRIVE testing. (a) Green channels of selected image and their 1st-manual images. (b) Local-phase-based enhancement after applying Retinex to green channel. (c) Wavelet-based enhancement method. (d) Multiscale multiorientation BTH. (e) Multiscale BTH.

TABLE 4: Illustrative comparison between all possible combinations of inhomogeneity correction and illumination enhancement methods. 1st row: local phase-based method [5]. 2nd row: wavelet-based method [10]. 3rd/7th row: CLAHE correction [3, 4, 41] with multiscale multiorientation BTH/multiscale BTH. 4th/8th row: retinex correction [5] with multiscale multiorientation BTH/multiscale BTH. 5th/9th row: median blurring correction [6] with multiscale multiorientation BTH/multiscale BTH. 6th/10th row: low-pass Gaussian blurring correction with multiscale multiorientation BTH/multiscale BTH.

| Method | CII | PSNR | M_{SSIM} |
|--|--------------|--------------|------------|
| Phase-based | 0.412 | 42.46 | 0.797 |
| Wavelet-based | 0.701 | 42.76 | 0.018 |
| CLAHE + multiscale multiorientation BTH | 0.528 | 42.79 | 0.802 |
| Retinex + multiscale multiorientation BTH | 0.475 | 42.18 | 0.621 |
| Median + multiscale multiorientation BTH | 0.552 | 43.37 | 0.818 |
| Gaussian + multiscale multiorientation BTH | 0.554 | 44.18 | 0.818 |
| CLAHE + multiscale BTH | 0.679 | 44.28 | 0.816 |
| Retinex + multiscale BTH | 0.643 | 42.71 | 0.816 |
| Median + multiscale BTH | 0.669 | 44.59 | 0.818 |
| Gaussian + multiscale BTH | 0.714 | 44.63 | 0.818 |

and STARE, as depicted in Table 5. ARIVE dataset is not used because only a few studies used ARIVE dataset. The comparison shows that the proposed method is comparable to most of blood vessel detection frameworks. The slightly lower value is due to the fact that the proposed segmentation shows smaller and thinner vascular structures which are not generally annotated in the ground truth images. This can be clearly seen in Figure 8 which shows examples from STARE and ARIA datasets where the proposed segmentation method succeeds in detecting small and thin vessel-like patterns. It is also due to overestimated vessel width, which is a result

of initial spectral segmentation (Section 3.2.1), ignoring all spatial properties to build initial regions.

5. Discussion and Conclusion

In this paper, we proposed a new method to detect blood vessels in fundus images, which is based on three main steps: filtering-based correction, morphological-illumination enhancement, and graph-based segmentation.

Low-pass Gaussian blurring (filtering-based correction) is suggested to remove most noisy areas in retinal images, that

TABLE 5: Performance of segmentation methods based on sensitivity (Se), specificity (Sp), accuracy (Acc), and area under curve (AUC) on DRIVE and STARE datasets. The hand segmented images from first-manual observers are used as benchmarks (1st STARE manual is selected because all works used it).

| Methods | DRIVE | | | STARE | | |
|------------------------------|-------------|-------------|--------------|-------------|-------------|--------------|
| | Sensitivity | Specificity | Accuracy | Sensitivity | Specificity | Accuracy |
| Bankhead et al. [10] | 0.703 | 0.028 | 0.9371 | — | — | — |
| Estrada et al. [19] | — | — | 0.934 | — | — | — |
| Wang et al. [11] | — | — | 0.946 | — | — | 0.952 |
| Nguyen et al. [42] | — | — | — | — | — | — |
| Orlando and Blaschko [43] | 0.785 | 0.967 | — | — | — | — |
| Salazar-Gonzalez et al. [44] | 0.7512 | 0.0316 | 0.941 | 0.789 | 0.037 | 0.944 |
| Mithun et al. [18] | 0.471 | 0.970 | 0.936 | — | — | — |
| Azzopardi et al. [4] | 0.744 | 0.978 | 0.953 | 0.786 | 0.975 | 0.951 |
| Zhao et al. [5] | 0.744 | 0.978 | 0.953 | 0.786 | 0.975 | 0.951 |
| Imani et al. [6] | 0.752 | 0.975 | 0.952 | 0.750 | 0.975 | 0.959 |
| <i>Proposed method</i> | 0.850 | 0.944 | 0.934 | 0.633 | 0.950 | 0.924 |

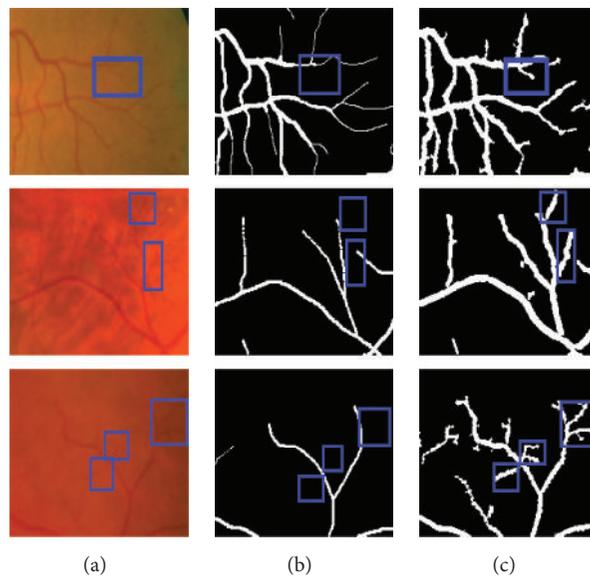


FIGURE 8: Illustrative comparison between results obtained from manual observers and the proposed segmentation on selected sample from STARE and ARIA datasets. (a) Selected sample. (b) 2nd manual sample (selected because it presents most small/thin blood vessels). (c) Segmented sample.

is, optical disk and fovea. This correction method may not be the best method; however, it has succeeded in detecting the most noisy areas from retinal images.

Due to high sensitivity of multistructure elements to edges in all directions, multiscale/multiorientation and multiscale morphological BTHs (morphological-illumination enhancement) are proposed which are capable of detecting most of the blood vessel edges which are thin and small. The deficiency of missing some thin vessels is due to the minimum threshold of connected component labeling of BTH-layer. Therefore, the need for appropriate thresholding method to find thin vessel and avoid false-edge pixel is important. Hence, one of our future works is to find more suitable connected component thresholding method to increase the accuracy of the proposed method.

Hierarchical graph-based segmentation is based on applying perceptual Gestalt principles (e.g., similarity, closure, proximity, and continuity) of spectral/spatial features between nodes in graphical multilayers. For instance, similarity of spectral characteristics between adjacent nodes is used to group small nodes. These characteristics are informative; however, semantic ambiguity still exists because of the appearance, shape, or other higher-level features' similarity. Moreover, closure principle is applied by combining nodes if their connected component labels are not part of the common connected component label of BTH-layer. From authors' viewpoint, there are other laws behind this closure principle which are similar in size and similar in orientation because BTH operator preserves multistructure which has variety in size and orientation.

This work solved the problem of the priority of the highest importance principle, unlike other works [28]. The major law in this work is proximity (connectedness). This is because the adjacent nodes in each layer are taken into account to make vessel-like pattern stand out from its background as a separate object by grouping small subobjects together with their surroundings based on other principles.

The quantitative performance of enhancement as well as segmentation shows that the proposed method works well on healthy retinal image with accuracy 93.4%. However, a main drawback is that it tends to produce false detection in overlapping areas. This has lowered the overall accuracy of the proposed method, especially on STARE images. To solve this problem, two-label graph-cut step will help to reduce incorrect detection and improve performance of the proposed segmentation.

In this paper, the proposed method is objectively evaluated. From the view of cognitive psychology, subjective evaluation is an efficient method to test the performance of perceptual segmentation. Therefore, we aim to investigate human visual perception for segmenting and consequently detecting blood vessels (new ground truth) and comparing with segmentation results.

Algorithm run-time is an issue of concern to assess the algorithm performance. In this paper, the computation time of the proposed method is 10 minutes and 3 seconds. The preprocessing part takes around 3 s and segmentation part takes about 10 m to analyze graph algorithms. The proposed algorithm is quite slow because of sequential connected component labeling and Dijkstra algorithms in graph-based segmentation. Therefore, in the future, we aim to parallelize connected component labeling and Dijkstra algorithms by translating our graph-based segmentation into parallel segmentation using massively parallel GPU [49].

Competing Interests

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

References

- [1] S. J. Lee, C. A. McCarty, H. R. Taylor, and J. E. Keeffe, "Costs of mobile screening for diabetic retinopathy: a practical framework for rural populations," *The Australian Journal of Rural Health*, vol. 9, no. 4, pp. 186–192, 2001.
- [2] W. Tasman, A. Patz, J. A. McNamara, R. S. Kaiser, M. T. Trese, and B. T. Smith, "Retinopathy of prematurity: the life of a lifetime disease," *American Journal of Ophthalmology*, vol. 141, no. 1, pp. 167–174, 2006.
- [3] A. W. Setiawan, T. Mengko, O. Santoso, and A. Suksmono, "Color retinal image enhancement using clahe," in *Proceedings of the International Conference on ICT for Smart Society (ICISS '13)*, pp. 1–3, Jakarta, Indonesia, June 2013.
- [4] G. Azzopardi, N. Strisciuglio, M. Vento, and N. Petkov, "Trainable COSFIRE filters for vessel delineation with application to retinal images," *Medical Image Analysis*, vol. 19, no. 1, pp. 46–57, 2015.
- [5] Y. Zhao, Y. Liu, X. Wu, S. P. Harding, and Y. Zheng, "Retinal vessel segmentation: an efficient graph cut approach with retinex and local phase," *PLoS ONE*, vol. 10, no. 4, Article ID e0122332, 2015.
- [6] E. Imani, M. Javidi, and H.-R. Pourreza, "Improvement of retinal blood vessel detection using morphological component analysis," *Computer Methods and Programs in Biomedicine*, vol. 118, no. 3, pp. 263–279, 2015.
- [7] G. L  th  n, J. Jonasson, and M. Borga, "Blood vessel segmentation using multi-scale quadrature filtering," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 762–767, 2010.
- [8] M. S. Miri and A. Mahloojifar, "Retinal image analysis using curvelet transform and multistructure elements morphology by reconstruction," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 5, pp. 1183–1192, 2011.
- [9] M. M. Fraz, S. A. Barman, P. Remagnino et al., "An approach to localize the retinal blood vessels using bit planes and centerline detection," *Computer Methods and Programs in Biomedicine*, vol. 108, no. 2, pp. 600–616, 2012.
- [10] P. Bankhead, C. N. Scholfield, J. G. McGeown, and T. M. Curtis, "Fast retinal vessel detection and measurement using wavelets and edge location refinement," *PLoS ONE*, vol. 7, no. 3, Article ID e32435, 2012.
- [11] C. Wang, N. Komodakis, and N. Paragios, "Markov Random Field modeling, inference and learning in computer vision and image understanding: a survey," *Computer Vision and Image Understanding*, vol. 117, no. 11, pp. 1610–1627, 2013.
- [12] M. Liao, Y.-Q. Zhao, X.-H. Wang, and P.-S. Dai, "Retinal vessel enhancement based on multi-scale top-hat transformation and histogram fitting stretching," *Optics and Laser Technology*, vol. 58, pp. 56–62, 2014.
- [13] A. Hoover, V. Kouznetsova, and M. Goldbaum, "Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response," *IEEE Transactions on Medical Imaging*, vol. 19, no. 3, pp. 203–210, 2000.
- [14] T. Kauppi, V. Kalesnykiene, J.-K. Kamarainen et al., "The DIARETDB1 diabetic retinopathy database and evaluation protocol," in *Proceedings of the 18th British Machine Vision Conference (BMVC '07)*, September 2007.
- [15] M. Niemeijer, B. van Ginneken, M. J. Cree et al., "Retinopathy online challenge: automatic detection of microaneurysms in digital color fundus photographs," *IEEE Transactions on Medical Imaging*, vol. 29, no. 1, pp. 185–195, 2010.
- [16] D. Mar  n, A. Aquino, M. E. Geg  ndez-Arias, and J. M. Bravo, "A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariants-based features," *IEEE Transactions on Medical Imaging*, vol. 30, no. 1, pp. 146–158, 2011.
- [17] M. M. Fraz, P. Remagnino, A. Hoppe et al., "Blood vessel segmentation methodologies in retinal images—a survey," *Computer Methods and Programs in Biomedicine*, vol. 108, no. 1, pp. 407–433, 2012.
- [18] N. C. Mithun, S. Das, and S. A. Fattah, "Automated detection of optic disc and blood vessel in retinal image using morphological, edge detection and feature extraction technique," in *Proceedings of the 16th International Conference on Computer and Information Technology (ICCIT '13)*, pp. 98–102, Khulna, Bangladesh, March 2014.
- [19] R. Estrada, C. Tomasi, M. T. Cabrera, D. K. Wallace, S. F. Freedman, and S. Farsi, "Exploratory dijkstra forest based automatic vessel segmentation: applications in video indirect

- ophthalmoscopy (VIO),” *Biomedical Optics Express*, vol. 3, no. 2, pp. 327–339, 2012.
- [20] Y. Yin, M. Adel, and S. Bourennane, “Retinal vessel segmentation using a probabilistic tracking method,” *Pattern Recognition*, vol. 45, no. 4, pp. 1235–1244, 2012.
- [21] F. Zana and J.-C. Klein, “Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation,” *IEEE Transactions on Image Processing*, vol. 10, no. 7, pp. 1010–1019, 2001.
- [22] S. M. Zabihi, M. Delgir, and H. R. Pourreza, “Retinal vessel segmentation using color image morphology and local binary patterns,” in *Proceedings of the 6th Iranian Conference on Machine Vision and Image Processing (MVIP '10)*, pp. 1–5, Isfahan, Iran, October 2010.
- [23] B. Dashtbozorg, A. M. Mendonça, and A. Campilho, “An automatic graph-based approach for artery/vein classification in retinal images,” *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1073–1083, 2014.
- [24] K. Koffka, *Principles of Gestalt Psychology*, Harcourt, New York, NY, USA, 1935.
- [25] M. Wertheimer, “Laws of organization in perceptual forms,” in *Psychologische Forschung*, pp. 71–88, 1938.
- [26] J. Wagemans, J. H. Elder, M. Kubovy et al., “A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure-ground organization,” *Psychological Bulletin*, vol. 138, no. 6, pp. 1172–1217, 2012.
- [27] A. Richtsfeld, M. Zillich, and M. Vincze, “Implementation of Gestalt principles for object segmentation,” in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR '12)*, pp. 1330–1333, Tsukuba, Japan, November 2012.
- [28] R. G. Mesquita and C. A. B. Mello, “Segmentation of natural scenes based on visual attention and gestalt grouping laws,” in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '13)*, pp. 4237–4242, Manchester, UK, October 2013.
- [29] D. Pei, Z. Li, R. Ji, and F. Sun, “Efficient semantic image segmentation with multi-class ranking prior,” *Computer Vision and Image Understanding*, vol. 120, pp. 81–90, 2014.
- [30] P. Soille, *Morphological Image Analysis: Principles and Applications*, Springer, New York, NY, USA, 2nd edition, 2003.
- [31] P. Soille, “Morphological image compositing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 673–683, 2006.
- [32] J. Cousty, L. Najman, and B. Perret, “Constructive links between some morphological hierarchies on edge-weighted graphs,” in *Mathematical Morphology and Its Applications to Signal and Image Processing: 11th International Symposium, ISMM 2013, Uppsala, Sweden, May 27–29, 2013. Proceedings*, vol. 7883 of *Lecture Notes in Computer Science*, pp. 86–97, Springer, Berlin, Germany, 2013.
- [33] M. Elad, “Retinex by two bilateral filters,” in *Proceedings of the 5th International Conference on Scale Space and PDE Methods in Computer Vision, Scale-Space*, vol. 3459, pp. 217–229, April 2005.
- [34] M. R. Khokher, A. Ghafoor, and A. M. Siddiqui, “Image segmentation using fuzzy rule based system and graph cuts,” in *Proceedings of the 12th International Conference on Control, Automation, Robotics and Vision (ICARCV '12)*, pp. 1148–1153, Guangzhou, China, December 2012.
- [35] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.
- [36] A. Rezvanifar and M. Khosravifard, “Including the size of regions in image segmentation by region-based graph,” *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 635–644, 2014.
- [37] M. B. Dillencourt, H. Samet, and M. Tamminen, “A general approach to connected-component labeling for arbitrary image representations,” *Journal of the ACM*, vol. 39, no. 2, pp. 253–280, 1992.
- [38] K. Suzuki, I. Horiba, and N. Sugie, “Linear-time connected-component labeling based on sequential local operations,” *Computer Vision and Image Understanding*, vol. 89, no. 1, pp. 1–23, 2003.
- [39] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Pearson Education, 2nd edition, 2003.
- [40] Trimble-Definiens Joint Press Release, *Trimble Acquires Definiens—Earth Sciences Business to Expand Its Geospatial Portfolio*, 2010.
- [41] M. H. A. Fadzil, H. A. Nugroho, H. Nugroho, and I. L. Iznita, “Contrast enhancement of retinal vasculature in digital fundus image,” in *Proceedings of the International Conference on Digital Image Processing*, pp. 137–141, IEEE, Bangkok, Thailand, March 2009.
- [42] U. T. V. Nguyen, A. Bhuiyan, L. A. F. Park, and K. Ramamohanarao, “An effective retinal blood vessel segmentation method using multi-scale line detection,” *Pattern Recognition*, vol. 46, no. 3, pp. 703–715, 2013.
- [43] J. I. Orlando and M. Blaschko, “Learning fully-connected CRFs for blood vessel segmentation in retinal images,” in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2014*, P. Golland, N. Hata, C. Barillot, J. Hornegger, and R. Howe, Eds., vol. 8673 of *Lecture Notes in Computer Science*, pp. 634–641, Springer, New York, NY, USA, 2014.
- [44] A. Salazar-Gonzalez, D. Kaba, Y. Li, and X. Liu, “Segmentation of the blood vessels and optic disk in retinal images,” *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 6, pp. 1874–1886, 2014.
- [45] S. N. Shajahan and R. C. Roy, “An improved retinal blood vessel segmentation algorithm based on multistructure elements morphology,” *International Journal of Computer Applications*, vol. 57, pp. 31–36, 2012.
- [46] J. George and S. P. Indu, “Fast adaptive anisotropic filtering for medical image enhancement,” in *Proceedings of the 8th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT '08)*, pp. 227–232, Sarajevo, Balkans, December 2008.
- [47] D. T. Larose, *Discovering Knowledge in Data: An Introduction to Data Mining*, John Wiley & Sons, New York, NY, USA, 2005.
- [48] A. Fathi and A. R. Naghsh-Nilchi, “Automatic wavelet-based retinal blood vessels segmentation and vessel diameter estimation,” *Biomedical Signal Processing and Control*, vol. 8, no. 1, pp. 71–80, 2013.
- [49] NVIDIA Corporation, *NVIDIA CUDA C programming guide. Version 3.2*, 2011.

Research Article

Improved Adaptive Vibe and the Application for Segmentation of Complex Background

Le Chang,¹ Zhenghua Liu,^{1,2} and Yan Ren^{1,3}

¹*School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China*

²*Science and Technology on Aircraft Control Laboratory, Beihang University, Beijing 100191, China*

³*School of Information Engineering, Inner Mongolia University of Science and Technology, Baotou 014010, China*

Correspondence should be addressed to Le Chang; cl.buaa@126.com

Received 12 January 2016; Revised 17 February 2016; Accepted 19 April 2016

Academic Editor: Daniel Zaldivar

Copyright © 2016 Le Chang 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 problems that basic Vibe algorithm cannot effectively eliminate the influence of background noise, follower shadow, and ghost under complex background effectively, an adaptive threshold algorithm, AdaVibe, based on the framework of basic Vibe is proposed. Aiming at the shortage of the basic algorithm, this paper puts forward some improvement measures in threshold setting, shadow eliminating, and ghost suppression. Firstly, judgment threshold takes adjustment with the changes of background. Secondly, a fast eliminating ghost algorithm depending on adaptive threshold is introduced. Finally, follower shadow is detected and inhibited effectively through the gray properties and texture characteristics. Experiments show that the proposed AdaVibe algorithm works well in complex environment without affecting computing speed and has stronger robustness and better adaptability than the basic algorithm. Meanwhile, the ghost and follower shadow can be absorbed quickly as well. Therefore, the accuracy of target detection is effectively improved.

1. Introduction

As the basis of the research on target tracking and behavior identification, moving target detection is one of the most popular directions in the field of computer vision [1–4]. Moving target detection algorithms can be divided into the following categories, frame difference method [5], optical flow method [6], and background subtraction method, which can further be divided into the Gaussian mixture model background subtraction method [7, 8], codebook model background subtraction method [9], pixel-level Vibe algorithm [10–12], and so on. Among the methods mentioned above, frame difference method with less computational complexity is easy to design but tends to misjudge slow moving target. Optical flow method has high accuracy, but it is not suitable for real-time detection because of large computation. The Gaussian mixture model with the slow parameter estimation has difficulty achieving real-time performance requirements. The fixed threshold value of codebook model can easily lead the code to diverge. Vibe algorithm can extract the target in the initial few frames, with good adaptability and real-time performance.

However, the basic Vibe algorithm also has a series of problems in practice. (1) The threshold set up by experience is a fixed value and it has no adaptive adjustment during the running process of the algorithm, so the result is not robust enough. (2) It is easy to create ghost and the suppression of the ghost is slow [13]. (3) Vibe algorithm does not deal with the follower shadow and directly recognizes the shadow as a part of the target [10]. The detection accuracy is reduced due to follower shadow. To solve these problems, this paper puts forward an improved algorithm combining adaptive algorithm with Vibe algorithm. A new suppression strategy of ghost that depends on the Vibe mode is introduced. At the same time, follower shadow is effectively detected through the gray properties and texture characteristics.

2. Fundamental Principle of Basic Vibe Algorithm

The Visual Background Extractor (Vibe) algorithm was proposed by Barnich and Van Droogenbroeck of Belgium in

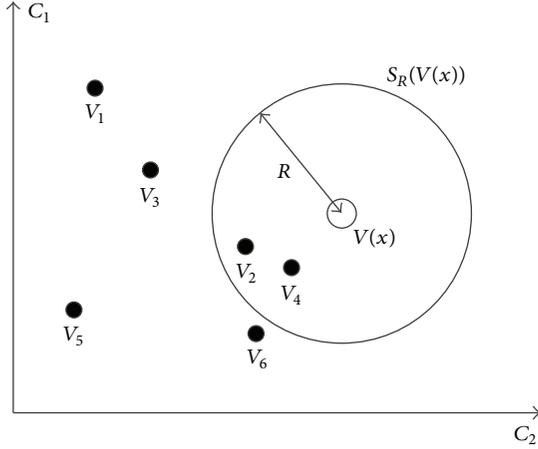


FIGURE 1: Vibe background model.

2009 for quick background subtraction and the moving target detection [14].

Vibe algorithm uses the first frame to initialize the background model. It introduces the randomization into modeling for the first time and proposes a strategy to randomly update the background model. At the same time, according to the spatial consistency principle, each pixel of the background will be randomly updated into the neighbor model. The steps of the algorithm are pixel background modeling, model matching, and model updating.

As shown in Figure 1, each background pixel x is modeled by a collection of N background sample values. The background model formula is shown as

$$M(x) = \{V_1, V_2, V_3, \dots, V_N\}. \quad (1)$$

All of the N samples have been judged as background pixels. As shown in Figure 2, these samples are randomly chosen from the eight pixels marked with k for N times.

In the next new frame, for the current pixel value $V(x)$ of x , set threshold R , and calculate the number of common samples of the $\{V(x) - R, V(x) + R\}$ and $M(x)$:

$$\# \{ \{V(x) - R, V(x) + R\} \cap M(x) \}. \quad (2)$$

If the result is greater than value $\#min$, x will be treated as a member of foreground. Otherwise, it will be treated as a member of background.

Usually, we need a certain length of video sequences to complete the initialization and build background model. However, Vibe algorithm only uses the first frame to initialize the background model. As one frame could not contain N samples of a specific pixel, according to the spatial consistency principle of neighbor pixels, we can fill the background model for N times with its neighbor pixels. These neighbor pixels are randomly chosen according to a uniform law:

$$M^0(x) = \{V^0(y \mid y \in N_G(x))\}. \quad (3)$$

In formula (3), 0 represents the initial time, and $N_G(x)$ represents the eight-neighbor pixels. Vibe adopts a conservative model update policy. Foreground pixels will never be

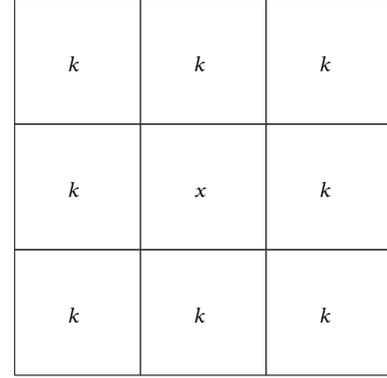


FIGURE 2: Eight-neighbor model.

used to fill the background model. When $V(x)$ is determined as a background pixel, it has a probability of $1/\phi$ to update its own background model. At the same time, it also has the probability of $1/\phi$ to update the background model of its neighbor pixels, which are selected randomly from its eight-neighbor pixels.

The present time is t ; in further time $(t + dt)$, the probability that a specific sample is still in the background model is equal to

$$P(t, t + dt) = e^{-\ln(N/(N-1))dt}. \quad (4)$$

3. Adaptive Threshold Strategy

In basic Vibe algorithm, model matching always uses a fixed global threshold R . Experiments show that the simple threshold policy ignores the complexity of the local environment and the uncertainty of changes. So it is difficult to detect the target effectively in complex environment.

For complex dynamic background, the threshold R should be increased appropriately, so that the background cannot be easily detected as foreground. On the other hand, for simple static background, R should be decreased to detect small changes of the foreground. In order to improve the robustness of basic Vibe algorithm, an improved method of adaptive threshold is proposed in this section.

Define a set of distances $D(x)$:

$$D(x) = \{D_1(x), \dots, D_i(x), \dots, D_N(x)\}. \quad (5)$$

In formula (5), $D_i(x)$ is the minimum Euclidean distance between current pixel $V(x)$ and its background samples $V_i(x)$ and N represents the number of samples. In this paper, N is set to be 20. The average value $d_{\text{mean}}(x)$ is used to characterize the complex situation of background change:

$$d_{\text{mean}}(x) = \sum_{i=1}^N D_i(x). \quad (6)$$

A new $d_{\text{mean}}(x)$ is recorded after each successful match between $V(x)$ and $V_i(x)$. For static background, $d_{\text{mean}}(x)$ tends to be steady, while $d_{\text{mean}}(x)$ increases gradually for

complex dynamic background. So, the threshold R is updated adaptively according to the value of $d_{\text{mean}}(x)$:

$$R = \begin{cases} R * (1 + \varepsilon_c), & \text{if } R < d_{\text{mean}}(x) * \delta, \\ R * (1 - \varepsilon_d), & \text{else.} \end{cases} \quad (7)$$

In formula (7), ε_c , ε_d , and δ are the fixed parameters; set $\varepsilon_c = 0.06$, $\varepsilon_d = 0.4$, and $\delta = 5$. It can be shown that R tends to be steady for static background. When the interference generated by dynamic background appears, R increases gradually to improve adaptability of the algorithm.

The upper and lower limits are set to prevent R from becoming too large or too small:

$$\begin{aligned} R_{\min} &= 10, \\ R_{\max} &= 30. \end{aligned} \quad (8)$$

4. The Adaptive Algorithm of Ghost Suppression

Different from other algorithms, Vibe algorithm initializes the background model from the first frame of the video, so the initialization is very quick. The drawback is that the presence of a moving object in the first frame will introduce an artifact commonly called a ghost. While initializing model, the moving target of the foreground in the first frame is considered as background pixels incorrectly. When the moving target in the following frames leaves the original position, the sampled real background grayscale values cannot match the initial background model. As a result, the background pixels are considered as foreground by mistake, and the ghost appears. Although in following frames the ghost can be eliminated by refreshing background models, this process is still relatively slow. Meanwhile, if the moving target goes through the ghost area, the detection accuracy will be decreased.

Considering the existence of the ghost, this paper deals with the detected foreground pixels according to properties of ghost to eliminate the ghost quickly.

4.1. Fundamental Idea. It can be seen from above that ghost pixels are background pixels, which are detected as foreground pixels incorrectly. According to the consistency principle of neighbor space, the grayscale values of ghost area are close to its neighbor pixels. So, a pixel can be treated as the ghost if this pixel and the background model of its neighbor pixels can match well. Usually, when we deal with a frame, we scan the frame from left to right and from top to bottom. As shown in Figure 3, following the left-right-top-bottom order, x stands for the suspicious ghost pixel to be judged, pixels marked with k are the background or treated ghost pixels, and pixels marked with u are the pixels to be dealt with.

When dealing with ghost, as the pixels marked with k_i have been scanned and dealt with, the background models of these four pixels are reliable. If pixel x is detected as a ghost pixel, update the background model of x in time. So background model of x is reliable and the ghost can be eliminated from outside to inside gradually to achieve the goal of quickly eliminating ghost. When updating the

| | | |
|-------|-------|-------|
| k_1 | k_2 | k_3 |
| k_4 | x | u |
| u | u | u |

FIGURE 3: Ghost processing pattern.

background model of x , we only choose its value from background models of four reliable neighbor pixels.

4.2. Detailed Process. Steps to deal with ghost are listed below.

- (1) Calculate the Euclidean distances in RGB space between x and pixels $\{I_1, I_2, I_3, I_4\}$ in positions $\{k_1, k_2, k_3, k_4\}$. Keep the distances as $\{d_1, d_2, d_3, d_4\}$, and get the minimum value of the four distances:

$$d_i = \sqrt{(r_i - r_x)^2 + (g_i - g_x)^2 + (b_i - b_x)^2}, \quad (9)$$

$$d_m = \min(d_1, d_2, d_3, d_4).$$

- (2) Using the corresponding pixel of d_m as reference pixel (i.e., $d_m = d_1$), calculate distances between the other three pixels and the reference. Find the largest distance D_m among the three distances following formula (10), and get the threshold by formula (11):

$$D_m = \max(D_{1,2}, D_{1,3}, D_{1,4}) \quad (10)$$

$$\text{th} = \min(D_m, D). \quad (11)$$

In formula (10), $D_{1,x}$ represents the distance between pixel on k_1 and pixels on $\{k_2, k_3, k_4\}$, respectively. It is to be noted that D in formula (11) is a fixed value to prevent the incidental fluctuations of threshold caused by occasional disturbances, and it is set to be 25 in this paper.

- (3) Use the threshold th to judge ghost pixels. When $d_m < \text{th}$, the pixel x can easily match the background model of its neighbor pixels, so x is considered to be a ghost pixel. Classify the pixel x as background, randomly sample the pixel again, and establish a new background model. The background models of its neighbor pixels are updated at the same time. Otherwise, the pixel x belongs to foreground target indeed.

To reduce the computation cost, not all detected foreground pixels are judged whether they are ghost using the

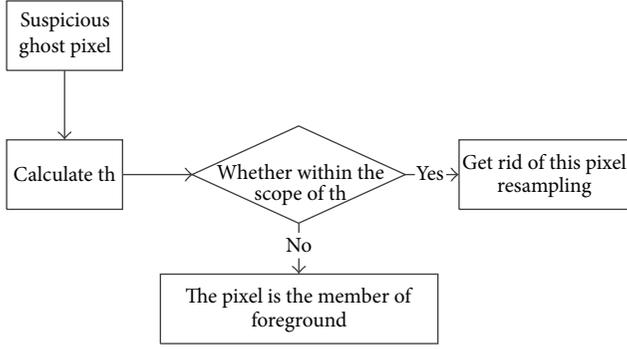


FIGURE 4: Flow chart of ghost suppression.

method above. Instead, a kind of classification strategy is used to filter the pixels before judging. In this paper, we count the times, N_g , that a pixel is continuously detected to be a foreground pixel. A pixel is determined to be a suspicious ghost pixel only when N_g exceeds a specific range. Then, take the method above to judge it again.

Figure 4 shows the flow chart of ghost suppression.

5. The Algorithm of Eliminating Shadow Based on Texture Characteristics

In general, shadow usually does not affect detection results. However, the follower shadow, which accompanies the target, is recognized as a part of the target by most algorithms. Basic Vibe algorithm does not take account of the follower shadow. That is to say, the follower shadow may be mixed into the detected foreground, which affects the result. To enhance the accuracy of the detected result, a new method of follower shadow eliminating is proposed.

Due to the characteristics of shadow, when a pixel is covered by shadow, its grayscale value is linear with that when not covered:

$$p(i, j) = \frac{B(i, j)}{O(i, j)} \quad (0.2 < p < 0.8). \quad (12)$$

According to the theory above, if the relationship between grayscale value of a foreground pixel and its original grayscale value satisfies formula (12), the pixel could locate in shadow area. In formula (12), $O(i, j)$ is the original grayscale value and $B(i, j)$ is the grayscale value when a pixel is covered by shadow. The whole formula is named proportion constraint formula.

Further experiments show that eliminating the shadow only with grayscale value may exaggerate the shadow area. That is, some parts of the target are also recognized as shadow. To solve this drawback, SILTP texture characteristics operator is used as additional constraint condition. The research shows that SILTP characteristics operator of a pixel cannot be changed a lot when covered by shadow.

Suppose the pixel location of a graph is (x_c, y_c) ; SILTP encoding is shown as

$$\text{SILRP}_{N,R}^\tau(x_c, y_c) = \bigoplus_{k=0}^{N-1} s_\tau(I_c, I_k). \quad (13)$$

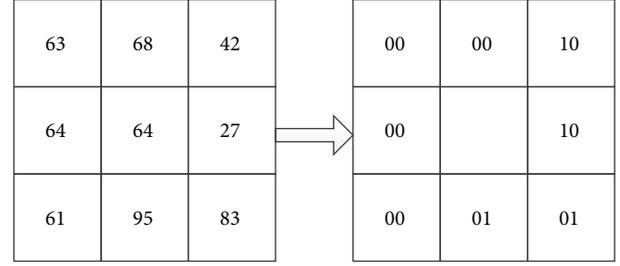


FIGURE 5: SILTP encoding.

In the formula, I_c is the grayscale value of the region center and I_k are the grayscale values of pixels in the N neighbor area whose radius is R . \bigoplus means connecting all binary values to strings. τ is the variable threshold range. Piecewise function of $s_\tau(I_c, I_k)$ is shown as

$$s_\tau(I_c, I_k) = \begin{cases} 01, & I_k > (1 + \tau) I_c, \\ 10, & I_k < (1 - \tau) I_c, \\ 00, & \text{other.} \end{cases} \quad (14)$$

In this paper, region R is taken as the eight-neighbor area of the center pixel, and τ is taken as 0.1. Figure 5 is an example of SILTP encoding.

Take out codes clockwise, and then the eight binary values generated by four 2-bit binary values of four directions (top, left, bottom, and right) can be recognized as the characteristics operator of this window's center pixel. The eight binary values are also the texture information of this window's center pixel. In Figure 5, the SILTP characteristics operator is 00100100. SILTP characteristics operator has stronger robustness against brightness changes. In addition, the weak shadow area can also be detected well:

$$N_s = \text{num}(\overrightarrow{o\text{SILTP}} - \overrightarrow{b\text{SILTP}}). \quad (15)$$

Take the foreground pixel which satisfies formula (12) as the foreground pixel to be detected. In formula (15), $\overrightarrow{o\text{SILTP}}$ stands for the original characteristics operator of the pixel, and $\overrightarrow{b\text{SILTP}}$ is the updated SILTP characteristic operator of the pixel. N_s is the number of positions whose $\overrightarrow{o\text{SILTP}}$ and $\overrightarrow{b\text{SILTP}}$ values are different among the eight positions:

$$N_s < N_{\max}. \quad (16)$$

Compare N_s and N_{\max} ; if formula (16) works, the pixel locates in follower shadow area. Set $N_{\max} = 5$ in this paper.

In summary, the whole process of eliminating shadow is shown as below.

- (1) Calculate initial characteristic operator $\overrightarrow{o\text{SILTP}}$.
- (2) Get foreground pixels by AdaVibe algorithm.
- (3) Test grayscale proportion on foreground pixels.
- (4) Test if the proportion satisfies formula (12). If it satisfies it, go to Step (5). Otherwise, the pixel is foreground.

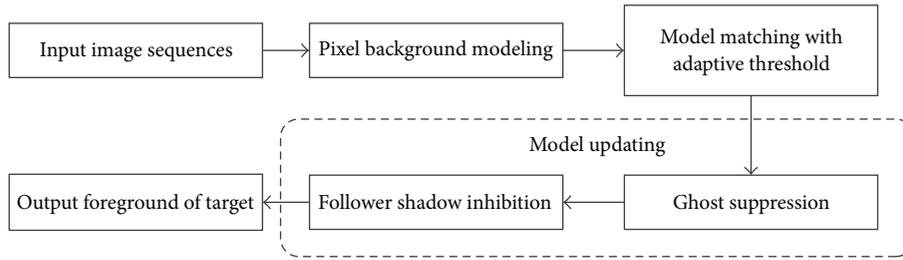


FIGURE 6: Flow chart of adaptive Vibe algorithm.

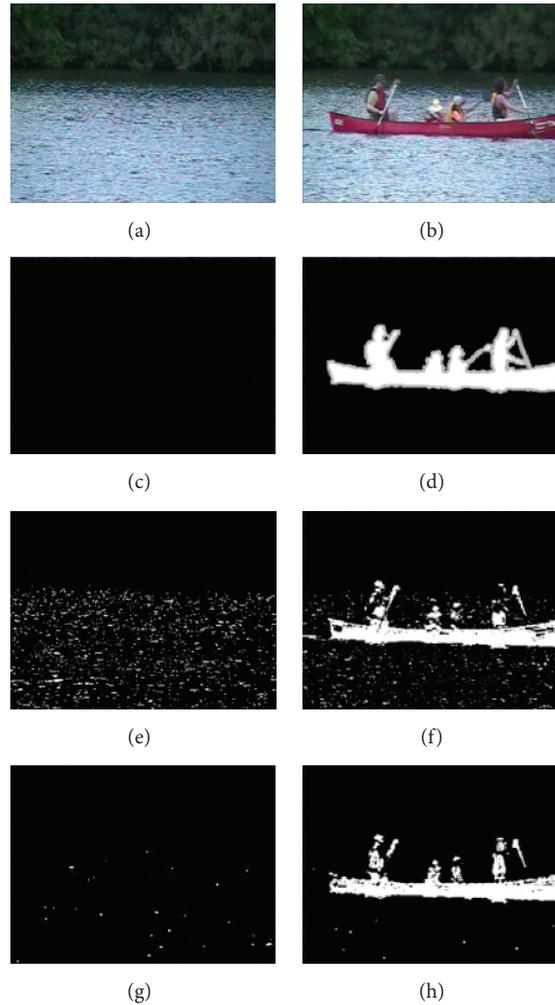


FIGURE 7: The effect figure by using Vibe and added adaptive R .

- (5) Calculate \overrightarrow{bSILTP} and N_s .
- (6) If N_s satisfies formula (16), the pixel is shadow and can be removed from the foreground. Otherwise, it is foreground.

If the video background varies greatly, to ensure the robustness of the algorithm, SILTP characteristic operators of background pixels should be refreshed every T cycles to create new \overrightarrow{oSILTP} . Usually, the foreground only occupies a small part of the whole area. Consequently, adding SILTP

characteristic operators does not affect the speed of the algorithm.

Above all, the flow chart of adaptive Vibe algorithm proposed in this paper is shown in Figure 6.

6. Simulations and Analyses

The experiment was done on a Lenovo PC, with Windows XP and Intel Celeron E3400 2.60 GHz CPU. The resolution of the video is $320 * 240$, and frame rate is 24 pfs, $\#min = 3$, $\phi = 16$.

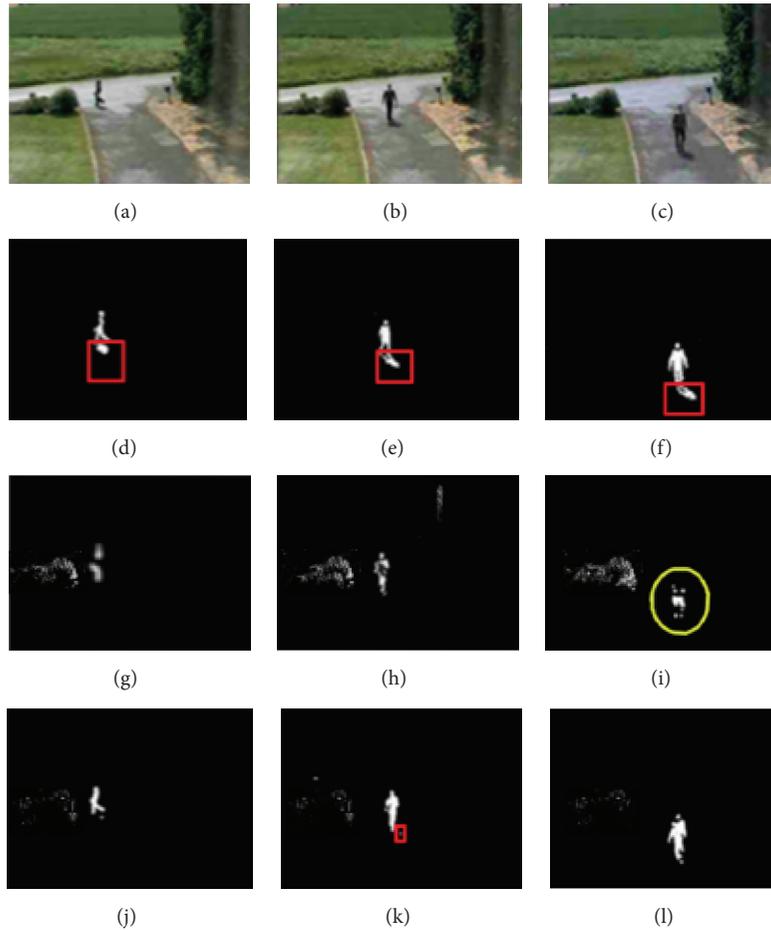


FIGURE 8: Effect figure of shadow eliminating.

TABLE 1: Performance of the algorithms.

| | Calculation speed/frames | Error rate/% |
|------------|--------------------------|--------------|
| Vibe (RGB) | 20.24 | 18.3 |
| AdaVibe | 19.68 | 11.2 |

Calculation speed: average number of frames treated per second.

Error rate: percentage of error against ground truth.

Vibe (RGB): Vibe algorithm in RGB space.

AdaVibe: AdaVibe algorithm proposed in this paper.

6.1. Test of Adaptive Threshold. Figure 7 is the effect diagram of the first test video clip. Left to right are frames 20 and 968, and top to bottom are original graph, ground-truth, Vibe (RGB), and AdaVibe proposed in this paper. In this video, the interference is water fluctuations.

Comparison between the two algorithms is shown as Table 1.

It can be seen that, with the adaptive threshold adjustment strategy, the influence of water ripple on the foreground detection is reduced. AdaVibe algorithm is better suitable for the dynamic background. So AdaVibe has stronger robustness and better adaptability than the basic algorithm. Besides, AdaVibe has the same speed as Vibe (RGB).

TABLE 2: Comparison of shadow eliminating.

| | Calculation speed/frames | Accuracy rate/% |
|------------|--------------------------|-----------------|
| Vibe (RGB) | 19.56 | N/A |
| SEGP | 19.07 | 68.4 |
| AdaVibe | 18.98 | 94.5 |

Accuracy rate: percentage of correct shadow pixels detected by algorithm against practical shadow pixels.

6.2. Test of Follower Shadow Elimination Added Adaptive Threshold. Figure 8 is the treated result of the second video clip. In this video, there is interference such as overlaps, vibrations of leaves, and light changes. Besides, there is a shadow accompanying the moving target. Left to right are frames 132, 296, and 333, and top to bottom are original graph, ground-truth, the shadow eliminating algorithm based on grayscale properties (SEGP), and AdaVibe. Area inside the red rectangle is the follower shadow.

Comparison of shadow eliminating among these algorithms is shown as in Table 2.

From Table 2, AdaVibe gains a good result. It can suppress the follower shadow, without lowering algorithm speed and oversuppression (marked by yellow circle in Figure 8).

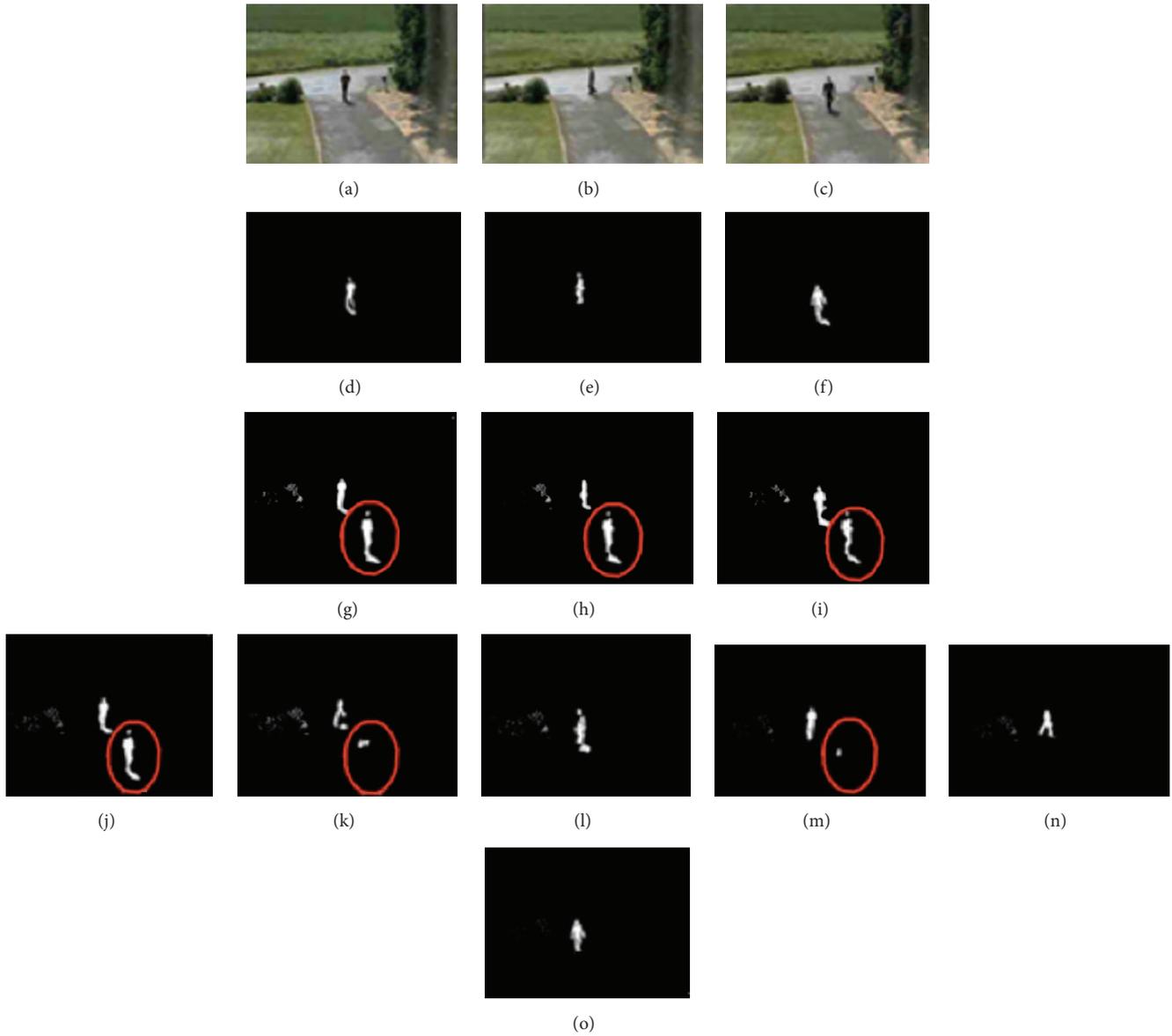


FIGURE 9: Effect figure of ghost and follower shadow suppression.

6.3. *Test of Ghost and Follower Shadow Elimination Added Adaptive Threshold.* In the third tested video, the moving target emerges from the first frame and there is follower shadow accompanying the target. That means ghost may exist in the basic Vibe algorithm. Apply different algorithms to the video, and compare the results.

Figure 9 is the effect diagram of ghost and follower shadow suppression. Left to right are frames 22, 117, and 303, and top to bottom are original graph, ground-truth, Vibe (RGB), algorithm proposed in [11], and AdaVibe. Area surrounded by red circle is ghost. The comparison of ghost suppression among three methods motioned above is listed in Table 3. Figure 10 shows the error rate of each frame.

From the test result, AdaVibe algorithm can eliminate ghost in less frames, which improves the computing speed.

What is more, elimination of ghost and shadow also improves the accuracy and robustness greatly.

7. Conclusion

To solve the drawbacks in basic Vibe algorithm, the adaptive threshold is proposed in this paper to cover the shortage of fixed threshold, reduce the interference with test results, and then take actions to eliminate ghost. Finally, grayscale and texture information are introduced to suppress follower shadow effectively.

The results show that AdaVibe algorithm can effectively eliminate background noise, follower shadow, and ghost in complex background and improve the accuracy and robustness as well as stableness without affecting the computing speed.

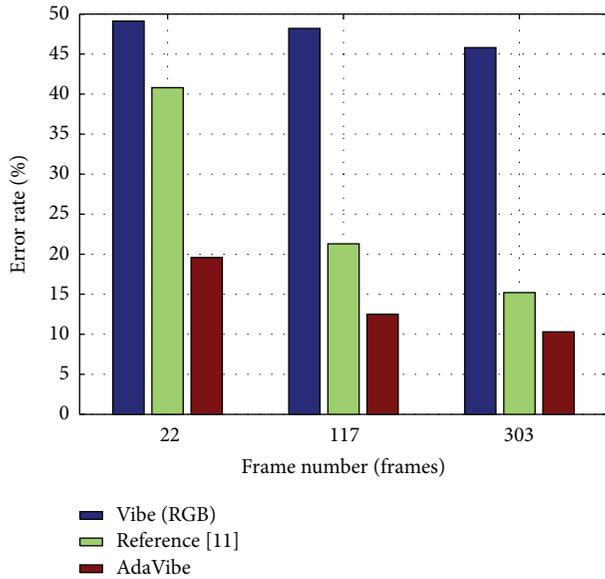


FIGURE 10: The error rate of each frame.

TABLE 3: Comparison of ghost inhibition.

| | Frames needed for ghost suppression | Calculation speed/frames |
|-------------------|-------------------------------------|--------------------------|
| Vibe (RGB) | 763 | 19.58 |
| Algorithm of [11] | 125 | 18.60 |
| AdaVibe | 28 | 18.83 |

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and thank the strong support provided by the National Natural Science Foundation of China (61563041), the National Natural Science Foundation of Inner Mongolia (2015MS0603), and Aviation Science Foundation Project (20135751040).

References

- [1] L. Zhang, F. Zhizhong, and Z. Yueping, "Moving objects detection based on HSV colorspace and Vibe algorithm," *Computer Engineering and Applications*, vol. 50, no. 4, pp. 181–185, 2014.
- [2] 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.
- [3] S. Cohen, "Background estimation as a labeling problem," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 2, pp. 1034–1041, Beijing, China, October 2005.
- [4] R. G. Abbott and L. R. Williams, "Multiple target tracking with lazy background subtraction and connected components

analysis," *Machine Vision and Applications*, vol. 20, no. 2, pp. 93–101, 2009.

- [5] J. Zhao, *The Research of Moving Target Detection Method Based on Three-frame Difference*, Xidian University, Xi'an, China, 2013.
- [6] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1–3, pp. 185–203, 1981.
- [7] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR '04)*, vol. 2, pp. 28–31, Cambridge, UK, August 2004.
- [8] 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.
- [9] C. Chen, X. Chen, and Z. Fan, "Detection algorithm based on block mode and codebook model," *Journal of China University of Metrology*, vol. 23, no. 2, pp. 125–130, 2012.
- [10] 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.
- [11] C. Liang, X. Chen, and Z. Fan, "Ghost suppression algorithm based on Vibe," *Journal of China University of Metrology*, vol. 24, no. 4, pp. 425–429, 2013.
- [12] 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*, pp. 32–37, IEEE, Providence, RI, USA, June 2012.
- [13] 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.
- [14] 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, Taipei, Taiwan, April 2009.

Research Article

Contrast Enhancement Method Based on Gray and Its Distance Double-Weighting Histogram Equalization for 3D CT Images of PCBs

Lei Zeng,^{1,2} Bin Yan,² and Weidong Wang¹

¹The General Hospital of People's Liberation Army, Beijing 100853, China

²National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China

Correspondence should be addressed to Bin Yan; ybspace@hotmail.com

Received 6 December 2015; Revised 13 March 2016; Accepted 7 April 2016

Academic Editor: Daniel Zaldivar

Copyright © 2016 Lei Zeng 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.

Cone beam computed tomography (CBCT) is a new detection method for 3D nondestructive testing of printed circuit boards (PCBs). However, the obtained 3D image of PCBs exhibits low contrast because of several factors, such as the occurrence of metal artifacts and beam hardening, during the process of CBCT imaging. Histogram equalization (HE) algorithms cannot effectively extend the gray difference between a substrate and a metal in 3D CT images of PCBs, and the reinforcing effects are insignificant. To address this shortcoming, this study proposes an image enhancement algorithm based on gray and its distance double-weighting HE. Considering the characteristics of 3D CT images of PCBs, the proposed algorithm uses gray and its distance double-weighting strategy to change the form of the original image histogram distribution, suppresses the grayscale of a nonmetallic substrate, and expands the grayscale of wires and other metals. The proposed algorithm also enhances the gray difference between a substrate and a metal and highlights metallic materials. The proposed algorithm can enhance the gray value of wires and other metals in 3D CT images of PCBs. It applies enhancement strategies of changing gray and its distance double-weighting mechanism to adapt to this particular purpose. The flexibility and advantages of the proposed algorithm are confirmed by analyses and experimental results.

1. Introduction

A printed circuit board (PCB) is an important part of electronic products. PCBs have played crucial roles in nearly every electronic product, such as electronic watches, cell phones, personal computers, digital machine tools, aerospace equipment, and medical instruments. Defects during the production process, as well as aging and wear during usage, will damage the entire board, thereby leading to abnormal operation, malfunction, and function invalidation in the circuit. These situations may eventually paralyze the entire working system. Therefore, monitoring the condition of PCBs is important. 3D imaging of PCBs via cone beam computed tomography (CBCT) can isotropically show the internal structure of PCBs in 3D space at micron-level resolution [1]. All information on internal defects, including cracks, incomplete welds, and welding faults, can be tested in a nondestructive manner [2].

PCBs consist of a nonmetallic substrate and metallic materials, such as wires, vias, pads, and copper. The metallic materials in PCBs provide the information of circuit connectivity. Therefore, we are concerned more about the metallic materials than about the nonmetallic materials. However, given the considerable amounts of metallic materials in PCBs, numerous factors, including the occurrence of metal artifacts and beam hardening, affect image quality in the CBCT imaging process [3, 4]. These harmful factors will result in low-contrast 3D CT images of PCBs. Thus, an image enhancement technology should be applied to highlight metallic materials. Histogram enhancement technology is the most basic image contrast enhancement method [5]. A histogram is the statistic probabilistic distribution of each gray level in a digital image. It can provide a general overview of the characteristics of an image, such as grayscale, gray level distribution and its density, the average luminance of an image, and image contrast [6]. Histogram enhancement can improve the gray

dynamic distribution of an image by changing the gray mapping relationship [7]. Histogram equalization (HE) is one of the most significant and commonly used histogram enhancement methods [8].

HE changes the input image into a new image with a uniform distribution of pixel gray values. It enhances the contrast of an image by increasing and adjusting the dynamic distribution range of gray. Given its simplicity and comparatively good performance in nearly all types of images, HE has become a popular image contrast enhancement technique [9]. However, HE also has several disadvantages. First, the HE algorithm uses global histogram information, which limits the intensity of image contrast stretching in some local areas. Consequently, this algorithm cannot effectively enhance the contrast between the background and a number of interesting details and has difficulty controlling the reinforcing effect. Second, saturation artifacts may occur in the result. Third, the grayscale of the output image may be overcombined, which may lead to the loss or discontinuity of the gray level; consequently, some detailed information of the image is lost.

To address these problems, several researchers [10–12] have enhanced images in subgray space. This strategy partially relieves the issues encountered in traditional HE and retains the details of an image as much as possible. These methods first separate the original global gray histogram into several subspaces. Then, the original image is divided into several subimages, and HE enhancement is performed on each subimage. Finally, all subimages are merged into a single image on the global grayscale histogram. The improved algorithms include brightness-preserving bi-HE (BBHE) [13, 14], recursive mean-separate HE (RMSHE) [15–17], dualistic subimage HE (DSIHE) [18, 19], minimum mean brightness error bi-HE (MMBEBHE) [20, 21], and weighting mean-separated sub-HE (WMSHE) [22, 23].

In BBHE, the input image histogram is divided into two subimages according to the mean value of the input. Then, both parts are merged again after the histogram is equalized. The result maintains the mean of the original image. The idea of preserving brightness is also adopted in the literature [24–27].

RMSHE follows the principle of BBHE. It iteratively divides the original image histogram into two parts at mean points. Then, 2^r subspaces are obtained after r partition times. HE is applied on each part. Evidently, as the number of partition times for the image histogram increases, the obtained result becomes increasingly closer to the original image. RMSHE is the same as BBHE if $r = 1$.

DSIHE selects the partition points of an image histogram, in which both parts have equal proportion. That is, both subimages have the same number of pixels after DSIHE partition.

MMBEBHE follows the basic principle of BBHE and DSIHE in decomposing an image and then applies HE to equalize the resulting subimages independently. However, MMBEBHE searches for a threshold T that decomposes the image I into two subimages, $I[0, T]$ and $I[T + 1, L - 1]$. Consequently, minimum brightness difference is achieved between the mean of the input image and that of

the output image. This threshold is essentially selected through enumeration.

WMSHE is similar to RMSHE. The only difference is that the former uses the gray-weighted mean as a partition rather than the original subimage. Compared with RMSHE, WMSHE can maintain more details of the image and reduce overenhancement.

Compared with the traditional HE algorithm, all the aforementioned algorithms can maintain more detailed information and can partially solve the overenhancement problem. However, these algorithms are not free from side effects [28].

Only two types of information are available in 3D CT images of PCBs. The first is nonmetallic materials, such as substrates, and the second is metallic materials, such as tracks, vias, and pads. However, various factors in the imaging process easily lead to the characteristic of a single peak in the image histogram. The gray of nonmetallic materials is smaller than that of metallic materials; thus, the gray of nonmetallic materials typically appears on the left side of the peak, whereas the gray of metallic materials appears on the opposite side. Metallic materials that provide electrical and signal connections for PCBs are of utmost concern. Hence, a good strategy should compress the grayscale of nonmetallic materials on the left side of the peak and extend the grayscale of the metallic materials on the right side. Such strategy enhances PCB images. However, traditional HE and its improved algorithms cannot change the gray probability density function (PDF) of the original image. They can only enhance an image by using the cumulative density function (CDF) of its gray probability density. Gray PDF or histogram determines the final enhancement directly. To solve this problem, several researchers have proposed to change the gray histogram of an original image before enhancing it. The representative algorithms include recursively separated and weighted HE (RSWHE) [29] and weighted thresholded HE (WTHE) [30, 31]. RSWHE achieves weighted HE by applying the strategy that considers the mean value or a value of equal volume as the partition point. It changes the original histogram based on the power of the sum of gray probabilities in each subsection. This strategy increases the gray of low probability in the new histogram. However, because of the insignificant capability of the algorithm to change the amplitude of the original histogram, the enhancement effect is only slightly improved. WTHE defines an upper limit of the maximum probability of gray and weakens the probability of the gray with high probability in the original histogram. By using a power function, WTHE also increases the probability of the gray that has low probability in the original histogram. The main aim is to maintain the low probability gray and reduce the high probability gray, thereby avoiding overenhancement. However, this algorithm is slightly flexible. For all image histograms, the algorithm only blindly reduces “high probability” and increases “low probability.” By contrast, maintaining the grayscale range with high probability is necessary to enhance PCB images because of the existence of PCB metals in this zone. Accordingly, this study presents an algorithm called image enhancement based on gray and its distance double-weighting HE (GDDWHE). As the name

suggests, GDDWHE introduces gray and its distance double-weighting mechanism to change the original gray distribution of the histogram, thereby satisfying the strategies and objectives of specific enhancements, as well as improving target enhancement.

2. HE Algorithm

The HE algorithm has been extensively used because it provides simple and fast calculation. This algorithm processes an input image to ensure that the gray histogram of the output image follows a uniform distribution. It increases and adjusts the distribution of the gray dynamic range to enhance the contrast of the image, thereby achieving image enhancement.

Assume that a 3D image $I(x, y, z)$ has N voxels and L gray levels $\{I_0, I_1, \dots, I_{L-1}\}$. The number n_k represents the number of voxels with a gray level of I_k . The occurrence probability of the gray level I_k is

$$p(I_k) = \frac{n_k}{N}, \quad (1)$$

where $k = 0, 1, \dots, L-1$. $p(I_k)$ is also known as the PDF, which defines the CDF

$$C(I_k) = \sum_{i=0}^k p(I_i), \quad (2)$$

where $k = 0, 1, \dots, L-1$. And $C(I_{L-1}) = 1$ is obtained by definition.

The main objective of the HE algorithm is to change the original image histogram from a concentrated distribution to a uniform distribution over the entire grayscale range. The HE algorithm uses CDF as a transfer mapping function. The grayscale of the original image is mapped in the entire grayscale dynamic range to achieve a uniform distribution. The transfer mapping function is achieved as follows:

$$f(I_k) = I_0 + (I_{L-1} - I_0)C(I_k). \quad (3)$$

If $\tilde{I} = \{\tilde{I}(x, y, z)\}$ is defined as the image after HE, then

$$\tilde{I} = f(I) = \{f(I(x, y, z)) \mid \forall I(x, y, z) \in I\}. \quad (4)$$

3. GDDWHE Algorithm

3.1. Design and Analysis of the Algorithm. In accordance with the characteristic of 3D CT images of PCBs, this paper proposes an image enhancement algorithm based on GDDWHE. The proposed algorithm focuses on the feature of a 3D image, modifies the probability distribution of the grayscale in the histogram that has a single peak, and obtains an advantageous enhancement effect. In the HE algorithm, the CDF is defined by (2). In this paper, however, a new cumulative density function $C(I_k)$ is defined as follows:

$$C(I_k) = \sum_{i=0}^k \left(\frac{I_i^\alpha \times \sum_{j=0}^i (p(I_j) \times d(I_i, I_j))}{\text{CONT}} \right), \quad (5)$$

where $\text{CONT} = \sum_{i=0}^{L-1} (I_i^\alpha \times \sum_{j=0}^i (p(I_j) \times d(I_i, I_j)))$ and is a constant for normalization to calculate the probability. $C(I_{L-1}) = 1$ is obtained by the upper definition. α is a parameter that controls the degree of the gray value I_i involved in the weighting process. $d(I_i, I_j)$ represents the distance between grayscales I_i and I_j . The subscripts i and j satisfy the following requirement: $0 \leq j \leq i \leq L-1$. I_j satisfies the following requirement: $I_0 \leq I_j \leq I_i$. If $I_i = I_j$, then $d(I_i, I_j) = 1$. In other cases, $0 < d(I_i, I_j) < 1$. A great distance between I_i and I_j corresponds to small $d(I_i, I_j)$. Similar to the summation in (2), the role of the first summation in (5) is to calculate the cumulative density function. As for the second summation in (5), we define the following expression:

$$\hat{p}(I_i) = \sum_{j=0}^i (p(I_j) \times d(I_i, I_j)). \quad (6)$$

Thus, (5) is transformed into

$$C(I_k) = \sum_{i=0}^k \frac{I_i^\alpha \times \hat{p}(I_i)}{\text{CONT}}. \quad (7)$$

In fact, (5) uses $(I_i^\alpha \times \hat{p}(I_i))/\text{CONT}$ to replace $p(I_i)$ in (2). The functions of $\hat{p}(I_i)$ and I_i^α are analyzed as follows: CONT is a constant. Thus, it is set aside temporarily in the following analyses.

In (2), $p(I_i)$ can be denoted by the following expressions:

$$\begin{aligned} p(I_i) &= 0 \times p(I_0) + 0 \times p(I_1) + \dots + 0 \times p(I_{i-1}) + 1 \\ &\times p(I_i) = \sum_{j=0}^i w_{ij} \times p(I_j), \end{aligned} \quad (8)$$

where the weight w_{ij} is denoted by the following expression:

$$w_{ij} = \begin{cases} 0, & j < i, \\ 1, & j = i. \end{cases} \quad (9)$$

The upper analysis indicates that $p(I_i)$ includes only the probability of gray level I_i and it does not consider the probability of gray level I_j when $j < i$ because the weight w_{ij} equals zero under this circumstance.

In (6), $\hat{p}(I_i)$ can be denoted by the following expressions:

$$\begin{aligned} \hat{p}(I_i) &= \sum_{j=0}^i (p(I_j) \times d(I_i, I_j)) \\ &= d(I_i, I_0) \times p(I_0) + d(I_i, I_1) \times p(I_1) + \dots \\ &\quad + d(I_i, I_{i-1}) \times p(I_{i-1}) + d(I_i, I_i) \times p(I_i) \\ &= \sum_{j=0}^i w_{ij} \times p(I_j), \end{aligned} \quad (10)$$

where the weight w_{ij} is denoted by the following expression:

$$w_{ij} = \begin{cases} d(I_i, I_j), & j < i, \\ d(I_i, I_j) = d(I_i, I_i) = 1, & j = i. \end{cases} \quad (11)$$

The distance $d(I_i, I_j)$ is used as the weight in the summation. The definition in (5) indicates that $d(I_i, I_j)$ is larger than zero. Unlike $p(I_i)$ in (2), $\hat{p}(I_i)$ in (6) uses weight $d(I_i, I_j)$ to replace zero when the subscript j is smaller than i . Thus, the summation in $\hat{p}(I_i)$ takes into account all the probabilities of the grayscale that is smaller than or equal to grayscale I_i . $\hat{p}(I_i)$ includes not only the probability of gray level I_i but also that of gray level I_j , which is smaller than I_i . In general, the weight will be small if the distance is far. Accordingly, the probability of the gray level close to I_i will have a larger contribution to $\hat{p}(I_i)$. The gray histogram of 3D CT images of PCBs usually has a single peak, and the histogram exhibits an increasing trend on the left side of the peak. The probability of the latter gray in particular will be higher than that of the former. The probability $p(I_j)$ is smaller than $p(I_i)$ when I_j is smaller than I_i . After being weighted by $d(I_i, I_j)$, which is smaller than 1, the contribution of $p(I_j)$ will decrease in the function $\hat{p}(x)$ compared with that in the function $p(x)$. Thus, (6) can compress the gray located on the left side of the peak. The gray histogram exhibits a decreasing trend on the right side of the peak. The probability of the latter gray is lower than that of the former. The probability $p(I_j)$ is larger than $p(I_i)$ when I_j is larger than I_i . All the weighted cumulative probabilities of the gray before I_i improve the final enhancement performance of the gray I_i in the function $\hat{p}(x)$ compared with that in the function $p(x)$. Thus, (6) can upgrade the gray located on the right side of the peak. The previous analysis indicates a low value of the gray on the left side of the peak in the histogram of the 3D image of a PCB; this low value corresponds to nonmetallic materials that should be suppressed in the enhancement result. The gray on the right side of the peak has a high value, which corresponds to metallic materials that should be enhanced in the enhancement result. The function $\hat{p}(x)$ can attain this goal through the preceding analysis of (6). Figure 1 shows the above analysis.

When $\alpha > 0$, I_i^α is a monotonically increasing function defined by I_i . When I_i increases, the product $I_i^\alpha \times \hat{p}(I_i)$ will enlarge $\hat{p}(I_i)$ in gray value I_i . That is, $I_i^\alpha \times \hat{p}(I_i)$ magnifies the large gray value more clearly than the small gray value. The preceding analysis indicates that, for 3D CT images of PCB histograms with a single peak, the controls of I_i^α can suppress the nonmetallic materials on the left side of the peak and enhance the metallic materials, such as wires, on its right side. Figure 5 shows the above analysis.

The proposed GDDWHE algorithm introduces a new cumulative density function using gray and its distance double-weighting strategy. The algorithm can flexibly change the original gray histogram or gray probability distribution into different forms by using different weighting mechanisms, such as selecting a different gray distance function and altering the value of α . For the gray histogram with a single peak, the algorithm can depress the histogram on the left side of the peak and raise the histogram on the right side of the peak. It is especially effective in enhancing the 3D CT images of PCBs that usually have a single peak in gray histograms. For the 3D CT images of PCBs, the proposed algorithm can compress the gray of nonmetallic substrates whose grays are

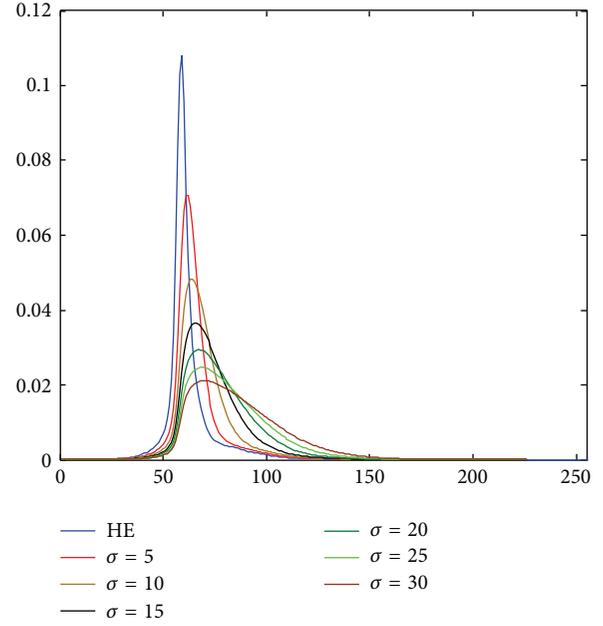


FIGURE 1: Modifications of the original histogram with different values of σ ($\alpha = 0$).

located on the left side of the peak and upgrade the gray of wires and metallic materials whose grays are located on the right side of the peak. The experimental results prove the advantage of the proposed algorithm.

The proposed algorithm proceeds as follows.

Step 1. The original image histogram $p(I)$ is calculated.

Step 2. The original histogram is weighted according to $I_i^\alpha \times \sum_{j=0}^i (p(I_j) \times d(I_i, I_j))$, and a new histogram and a new CDF are obtained.

Step 3. The new gray histogram is mapped according to $f(I_k) = I_0 + (I_{L-1} - I_0)C(I_k)$, and the final image enhancement results are obtained.

3.2. Discussion and Analysis of Parameter Selection. This section discusses the selection of different parameters of GDDWHE, including the distance function, the value of α , and their influences on the transformation of the gray histogram in a 3D CT image of a PCB with a single peak. Accordingly, we can select the optimum combination of parameters to enhance overall performance and achieve an improved enhancement result for 3D CT images of PCBs.

3.2.1. Selecting the Distance Function. Different distance functions $d(I_i, I_j)$ will obtain different enhancement effects. This study selects the squared Euclidean distance deformation form as follows:

$$d(I_i, I_j) = e^{-(I_i - I_j)^2 / 2\sigma^2}, \quad (12)$$

where σ is used to control the range of gray levels that participate in the operations.

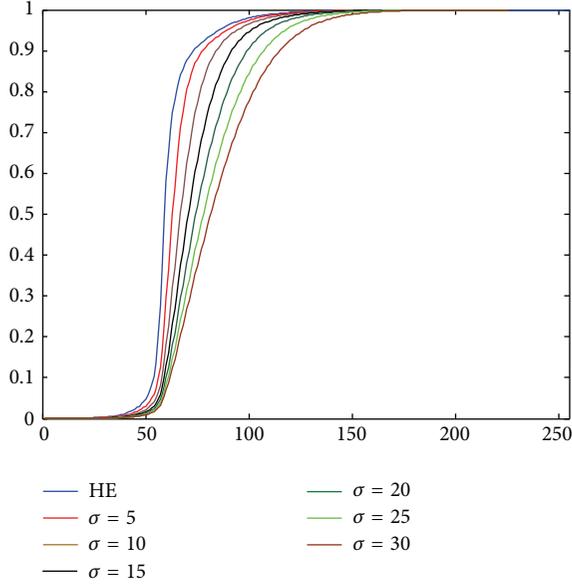


FIGURE 2: Modifications of the CDF of the original histogram with different values of σ ($\alpha = 0$).

3.2.2. Effects of Different Gray Variances on the Algorithm. The following analysis discusses the influence of different control parameters of the gray variance value σ on the enhancement effect. Figure 1 shows the modifications in the original histogram caused by different gray variances σ . To avoid the effect of the weighting of gray on the results, α is set to zero. The figure also shows that the single peak of the original histogram is widened and its amplitude decreases with the increase in σ . The decreasing trend on the right side of the peak and the increasing trend on the left side of the peak both slow down. Large values of σ indicate that the transformed peak is far from the original peak. Figure 2 shows the CDF generated by the changed gray histogram. This figure also indicates that the repression of the gray in the low gray value zone is highly evident with increasing σ , but the stretch of magnitude in the high gray value zone is weakened. This result is not conducive to the enhancement of metal PCBs. According to the preceding analysis, the gray values that correspond to wires and other metals will appear after the peak of the original histogram. Therefore, for 3D CT images of PCBs, large values of σ do not necessarily indicate good results. The stretching capability within the range of a large gray value should be considered. Evidently, small σ will improve the enhancement result. We choose $\sigma = 5$ on the basis of the actual gray value distribution feature of the image and the preceding comprehensive analysis. The selected experimental PCB image has two circuit layers, and the center position of each circuit layer is located on the 6th slice and the 43rd slice. Figure 3 shows the original images of both slices. Figure 4 shows the image of the 6th slice enhanced by different values of σ . We apply some statistical measurement indicators to assess the performance of the enhancement method. These indicators include the intensity distribution variance (δ^2), contrast (C), EME, Michelson law-based EME

(EME_Michelson), EME using entropy (EME_entropy), and AME [32–34].

The size of a 3D image $I(x, y, z)$ is $L \times W \times H$ and the image is broken up into $k_1 \times k_2 \times k_3$ nonoverlapped subblocks. $I_{\max;l,m,n}^w$ and $I_{\min;l,m,n}^w$ are the maximum and minimum of every subblock, respectively. In this paper, we set the subblock size to $3 \times 3 \times 3$ voxels. These aforementioned indicators are calculated by the following equations:

$$\delta^2 = \frac{1}{LWH} \sum_{x=1}^L \sum_{y=1}^W \sum_{z=1}^H (I(x, y, z) - \mu)^2, \quad (13)$$

where $\mu = (1/LWH) \sum_{x=1}^L \sum_{y=1}^W \sum_{z=1}^H I(x, y, z)$, which is the gray mean of the image;

$$C = \sum_{x=1}^L \sum_{y=1}^W \sum_{z=1}^H \sum_{\delta=0}^{255} \delta(x, y, z)^2 P_{\delta}(x, y, z), \quad (14)$$

where $\delta(x, y, z) = |I(x, y, z) - I(m, n, k)|$, $I(m, n, k)$ is one of the six neighborhoods of $I(x, y, z)$ in 3D space, and $P_{\delta}(x, y, z)$ is the probability of $\delta(x, y, z)$;

$$\text{EME} = \frac{1}{k_1 k_2 k_3} \sum_{l=1}^{k_1} \sum_{m=1}^{k_2} \sum_{n=1}^{k_3} 20 \ln \frac{I_{\max;l,m,n}^w}{I_{\min;l,m,n}^w + c}, \quad (15)$$

where c is a small constant equal to 0.000001 to avoid dividing by 0;

EME_Michelson

$$= \frac{1}{k_1 k_2 k_3} \sum_{l=1}^{k_1} \sum_{m=1}^{k_2} \sum_{n=1}^{k_3} 20 \ln \frac{I_{\max;l,m,n}^w - I_{\min;l,m,n}^w}{I_{\max;l,m,n}^w + I_{\min;l,m,n}^w + c}, \quad (16)$$

$$\text{EME_entropy} = \frac{\max}{\alpha} \{ \text{EME}_{k_1, k_2, k_3}(\alpha) \}, \quad (17)$$

where $\text{EME}_{k_1, k_2, k_3}(\alpha) = (1/k_1 k_2 k_3) \sum_{l=1}^{k_1} \sum_{m=1}^{k_2} \sum_{n=1}^{k_3} \alpha (I_{\max;l,m,n}^w / (I_{\min;l,m,n}^w + c))^{\alpha} \ln(I_{\max;l,m,n}^w / (I_{\min;l,m,n}^w + c))$;

$$\text{AME} = \frac{\max}{\alpha} \{ \text{AME}_{k_1, k_2, k_3}(\alpha) \}, \quad (18)$$

where $\text{AME}_{k_1, k_2, k_3}(\alpha) = (1/k_1 k_2 k_3) \sum_{l=1}^{k_1} \sum_{m=1}^{k_2} \sum_{n=1}^{k_3} \alpha ((I_{\max;l,m,n}^w - I_{\min;l,m,n}^w) / (I_{\max;l,m,n}^w + I_{\min;l,m,n}^w + c))^{\alpha} \ln((I_{\max;l,m,n}^w - I_{\min;l,m,n}^w) / (I_{\max;l,m,n}^w + I_{\min;l,m,n}^w + c))$.

In (17) and (18), α is set from 0.1 to 1 at intervals of 0.1.

The intensity distribution variance δ^2 and contrast C reflect the overall enhancement result of the image. Notably, large values indicate good overall enhancement. The other indicators, namely, EME, EME_Michelson, EME_entropy, and AME, reflect the enhancement effect on the local area of the image [32] or the capability to preserve local details. Large values indicate good capability to maintain local details. The indicators δ^2 and C have an order of priority that should be considered for their representations in the overall enhancement performance. The capability to stretch the gray in the local area is also considered. Table 1 presents the measurement indicators of the enhancement performance using different values of σ . We will select the appropriate parameter σ from Table 1. As shown in Table 1, improved enhancement is achieved when $\sigma = 5$.

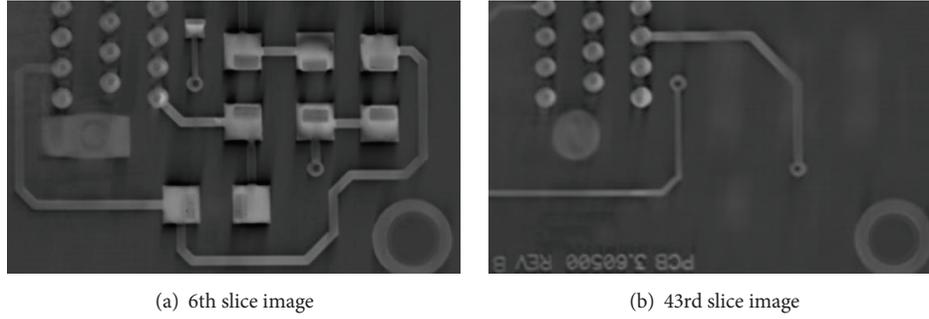


FIGURE 3: Two slice images of the original 3D CT image of the PCB.

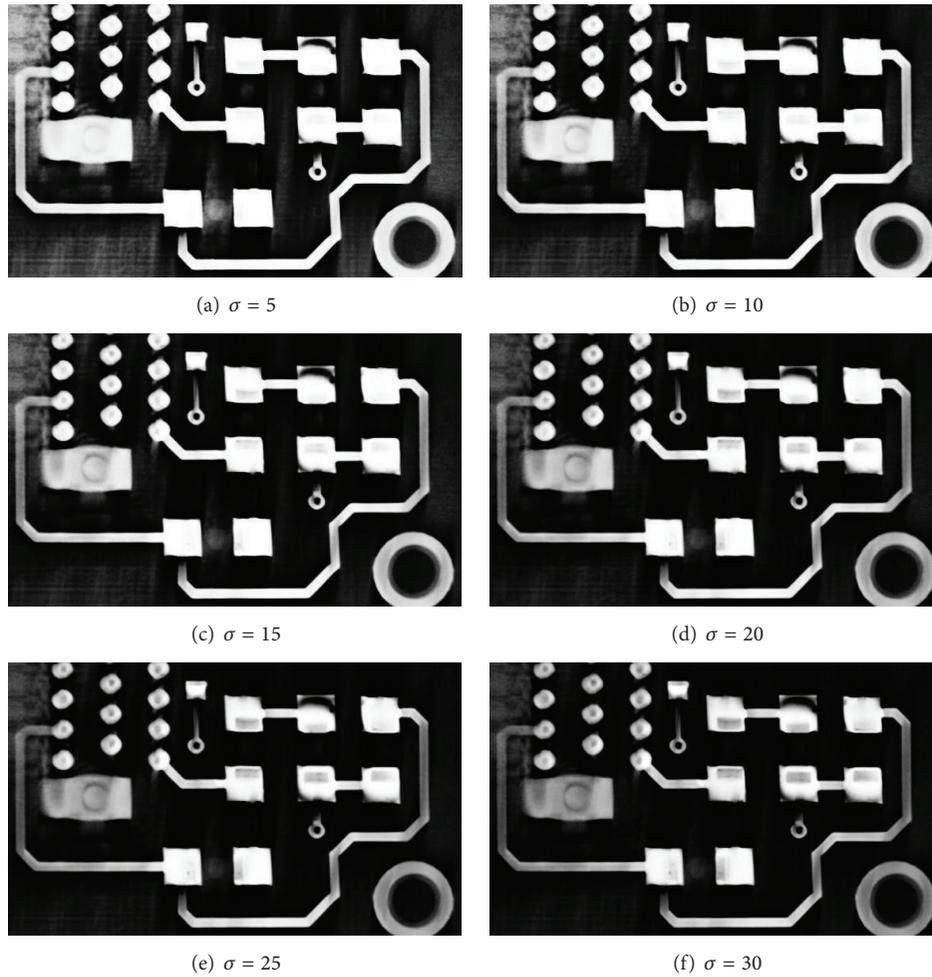


FIGURE 4: 6th slice images of the different enhancement results of the original 3D CT image of the PCB with different values of σ ($\alpha = 0$).

TABLE 1: Results of the enhancement indicators with different values of σ ($\alpha = 0$).

| | δ^2 | C | EME | EME_Michelson | EME_entropy | AME |
|---------------|------------|----------|---------|---------------|-------------|---------|
| $\sigma = 5$ | 5301.4 | 113.3032 | 48.6657 | -182.0618 | 1.6361 | -1.1349 |
| $\sigma = 10$ | 4476.2 | 78.8758 | 50.7792 | -171.4402 | 1.8711 | -1.1121 |
| $\sigma = 15$ | 3707.7 | 60.0761 | 52.3497 | -165.7646 | 2.0425 | -1.0985 |
| $\sigma = 20$ | 3060.1 | 47.6518 | 54.3363 | -161.2780 | 2.2999 | -1.0839 |
| $\sigma = 25$ | 2532.7 | 38.5262 | 56.0590 | -159.5976 | 2.6441 | -1.0795 |
| $\sigma = 30$ | 2114.4 | 31.7885 | 57.3516 | -157.5428 | 2.8432 | -1.0694 |

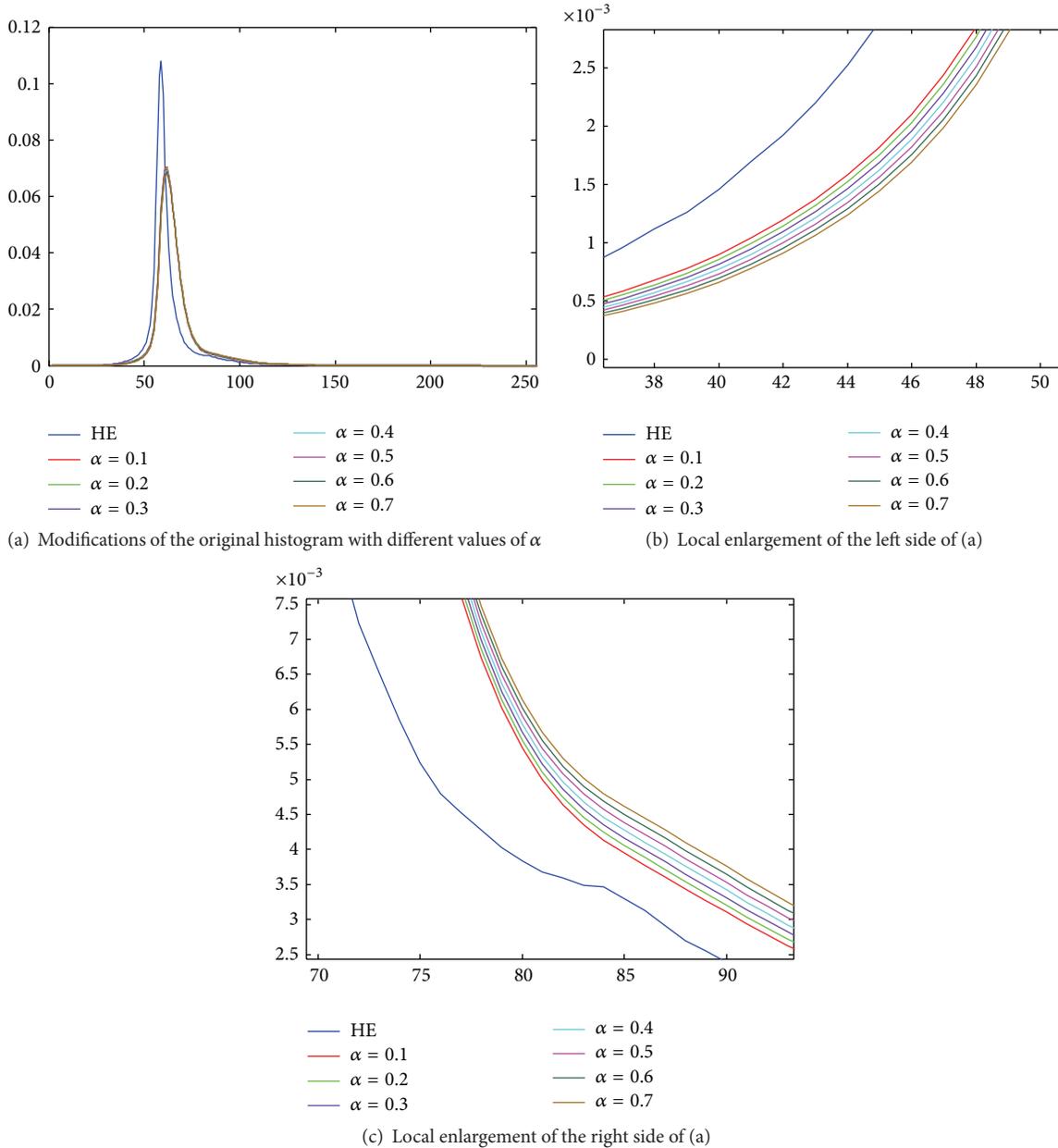


FIGURE 5: Modifications of the original histogram with different values of α ($\sigma = 5$).

3.2.3. *Effects of the Different Weights of the Gray Value on the Algorithm.* In the following analysis, we discuss the influence of the different values of the weight control parameter α on the enhancement result. On the basis of the preceding analysis, the parameter of the gray value variance σ is set to 5 ($\sigma = 5$). Figure 5 shows the modifications of the original histogram with different values of α . As shown in Figure 5(a), the results of different values of α are nearly the same. Figures 5(b) and 5(c) present the enlarged local histograms of the image in Figure 5(a). These enlarged histograms show that the results of different values of α are not highly evident. However, Figure 5(b) illustrates that the capability to depress the gray histogram increases within a small gray value range

with the increase in α . Figure 5(c) shows that the capability to upgrade the gray histogram increases within a large gray value range with the increase in α . Consequently, the overall effect will still improve. These results verify the previous analysis of the algorithm. Given the slight differences among the gray histogram results obtained using different values of α , we do not present the corresponding gray CDFs. Figure 6 presents the different enhancement results of the 43rd slice image obtained using different values of α . Table 2 provides the enhanced performance results of different values of α . We will select the appropriate parameter α from Table 2. On the basis of Figure 6 and Table 2, we select $\alpha = 0.5$ for the experimental 3D image when $\sigma = 5$.

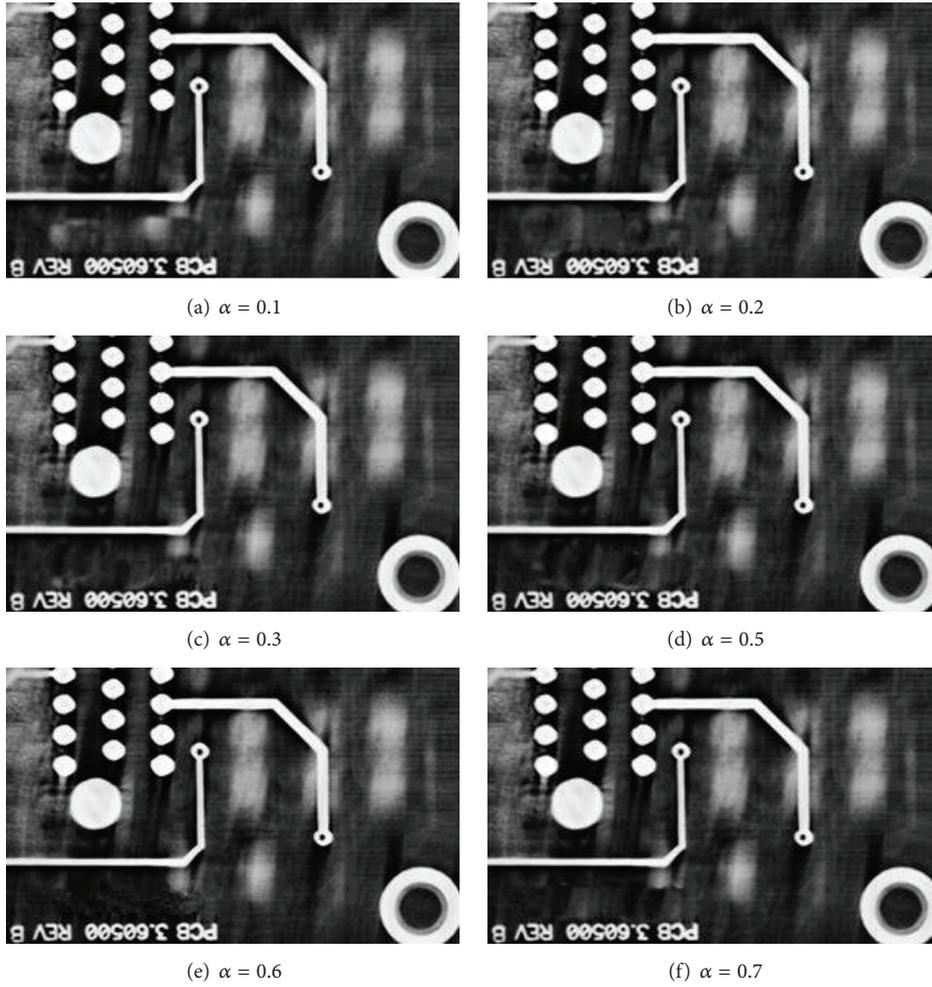


FIGURE 6: 43rd slice images of the different enhancement results of the original 3D CT image of the PCB with different values of α ($\sigma = 5$).

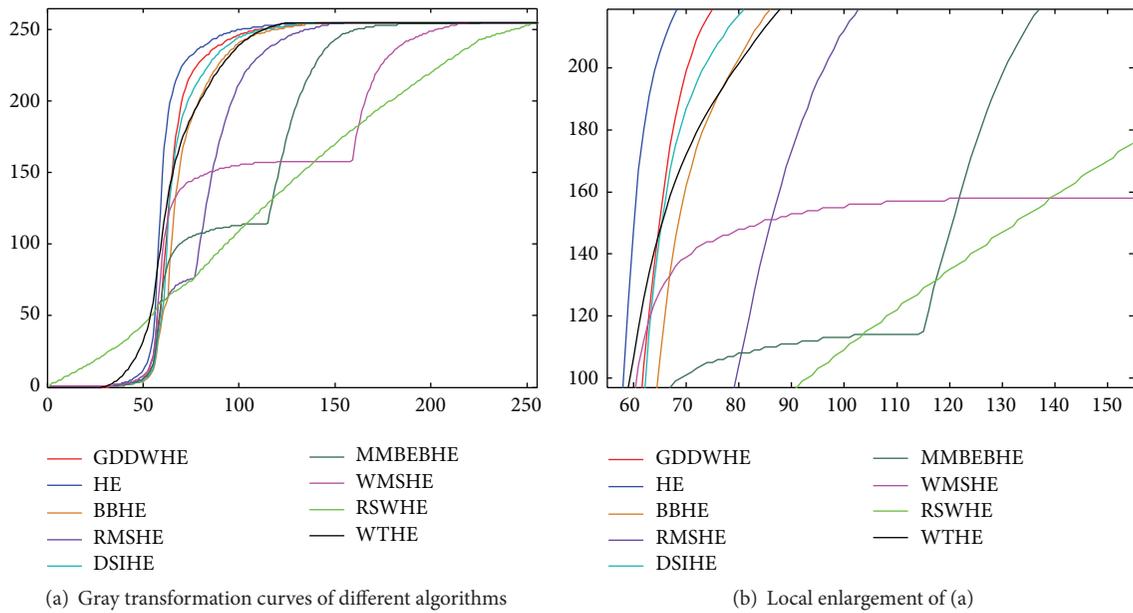


FIGURE 7: Gray transformation curves of different algorithms.

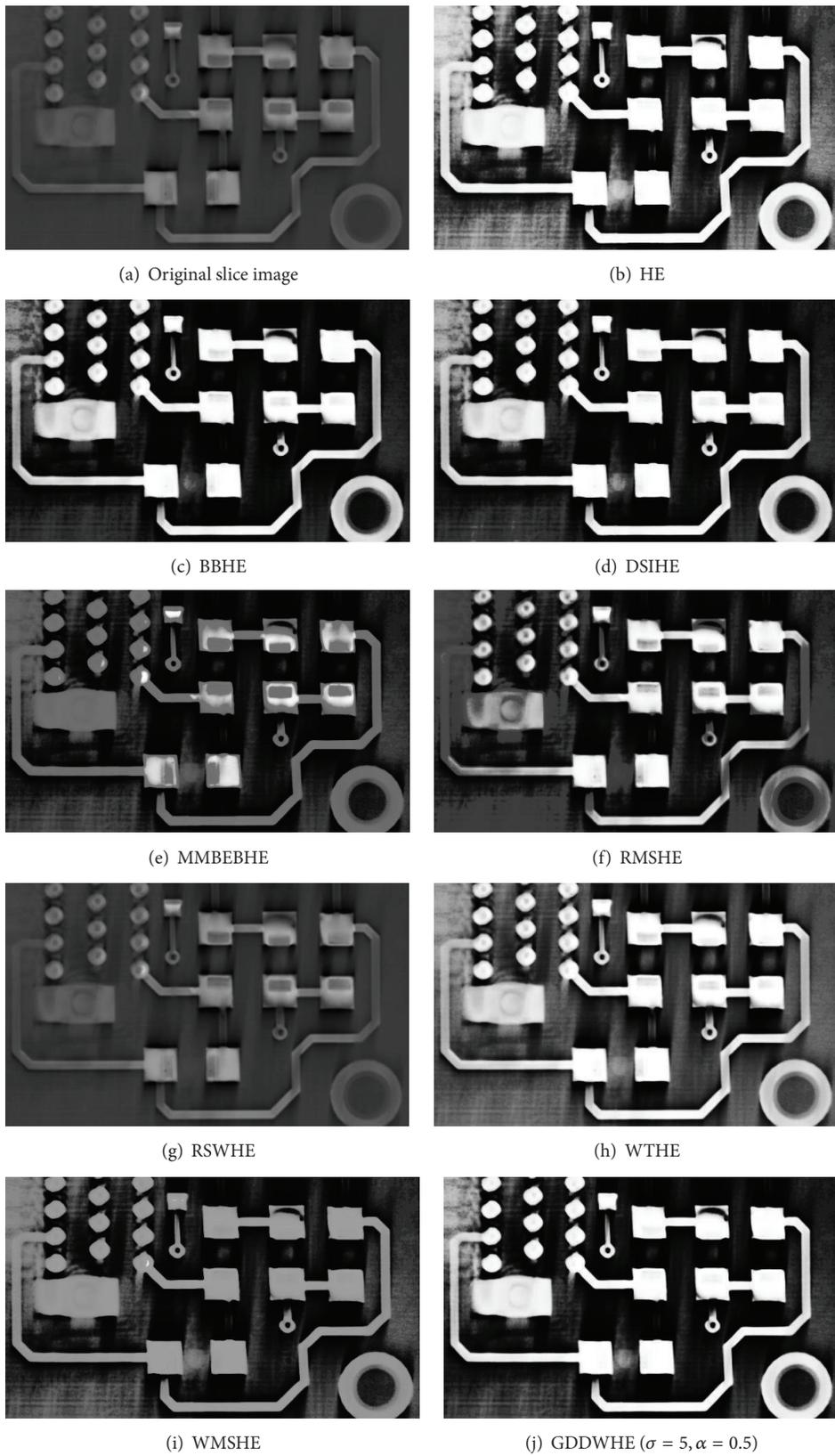


FIGURE 8: 6th slice images of the enhancement results of different algorithms for the original 3D CT image of the PCB.

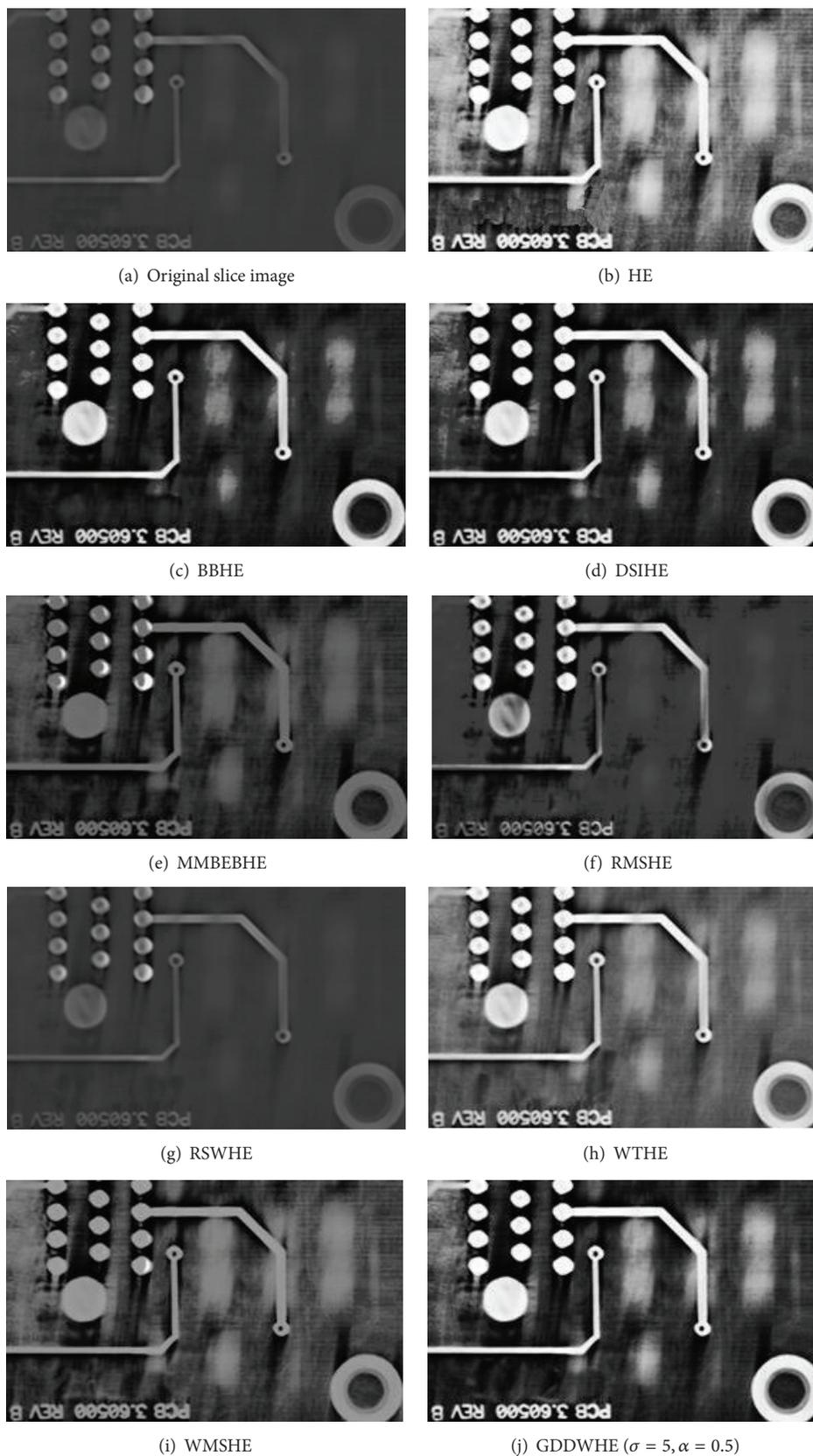


FIGURE 9: 43rd slice images of the enhancement results of different algorithms for the original 3D CT image of the PCB.

TABLE 2: Results of the enhancement indicators with different values of α ($\sigma = 5$).

| | δ^2 | C | EME | EME_Michelson | EME_entropy | AME |
|----------------|------------|----------|---------|---------------|-------------|---------|
| $\alpha = 0.1$ | 5300.1 | 113.1104 | 50.2641 | -180.2698 | 1.8186 | -1.1266 |
| $\alpha = 0.2$ | 5261.7 | 111.3227 | 50.8808 | -179.5898 | 1.9254 | -1.1247 |
| $\alpha = 0.3$ | 5238.4 | 110.4491 | 51.7547 | -178.4670 | 2.0297 | -1.1203 |
| $\alpha = 0.5$ | 5190.0 | 108.5125 | 53.8177 | -176.2185 | 2.3154 | -1.1113 |
| $\alpha = 0.6$ | 5165.7 | 107.6472 | 55.1504 | -174.9075 | 2.4976 | -1.1057 |
| $\alpha = 0.7$ | 5126.3 | 106.3441 | 55.2525 | -174.3375 | 2.4789 | -1.1041 |

TABLE 3: Results of the enhancement indicators of different algorithms.

| | δ^2 | C | EME | EME_Michelson | EME_entropy | AME |
|---------|------------|----------|---------|---------------|-------------|---------|
| BBHE | 4099.3 | 78.3485 | 44.5323 | -177.4295 | 0.5607 | -1.1293 |
| HE | 5186.8 | 159.0364 | 37.5031 | -215.8138 | 0.4681 | -1.2302 |
| DSIHE | 4992.2 | 107.1096 | 46.3229 | -175.5208 | 0.5863 | -1.1136 |
| MMBEBHE | 1178.6 | 35.4638 | 36.9988 | -211.7172 | 0.4566 | -1.2245 |
| RMSHE | 1581.6 | 44.8693 | 29.8655 | -253.1355 | 0.3690 | -1.2820 |
| WMSHE | 2002.2 | 61.5583 | 37.1405 | -214.1648 | 0.4613 | -1.2254 |
| RSWHE | 227.1 | 3.8843 | 8.0991 | -341.4916 | 0.0864 | -1.6056 |
| WTHE | 2587.0 | 49.9144 | 30.1331 | -235.8318 | 2.2706 | -1.3675 |
| GDDWHE | 5190.0 | 108.5125 | 53.8177 | -176.2185 | 2.3154 | -1.1113 |

3.3. Experimental Results and Analysis. In this section, the results of different enhancement algorithms for the selected 3D CT image of a PCB are tested and compared. The single peak of the gray histogram is located at the gray value equal to 59. The gray transformation curve reflects the ability to extend the contrast. Figure 7 shows the gray transformation curves of the different algorithms. Figure 7(b) demonstrates the local enlargement of the image in Figure 7(a). Figure 7(b) indicates that the proposed method stretches the gray rapidly when the gray value is larger than the peak in the histogram, and thus metals are enhanced more clearly. Figures 8 and 9 present the enhancement results of the 6th and 43rd slice images using different algorithms. Table 3 provides the enhancement performance results of different algorithms. Some parameters in the algorithms are set as follows: in GDDWHE, $\sigma = 5$, $\alpha = 0.5$; in WTHE, $\alpha = 0.5$; in RSWHE and RMSHE, $r = 2$; in WMSHE, $r = 1$. As shown in Table 3, the proposed method achieves the best performance among all the indicators. However, the indicators C and EME_Michelson are slightly smaller than those for HE and DSIHE, respectively. This finding indicates that the proposed method exhibits a strong advantage in enhancement and can achieve better enhancement results for 3D CT images of PCBs.

4. Summary

This study proposes an image enhancement algorithm based on GDDWHE according to the characteristics of 3D CT images of PCBs. The method changes the distribution form of the original histogram by using the gray and its distance double-weighting strategy. The gray of nonmetallic substrates is compressed in the histogram of 3D CT images of PCBs, whereas the gray of wires and metallic materials is elevated.

This approach further enhances the appearance of wires and other metals in 3D CT images of PCBs. Results of the algorithm analysis and the experiment indicate the effectiveness of the proposed method. Compared with other existing image enhancement algorithms, the proposed algorithm can more effectively extend the gray difference between substrates and metals in 3D CT images of PCBs and achieve better enhancement performance.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This study is supported by the National Natural Science Foundation of China under Grant no. 61372172 and the National High Technology Research and Development Program of China under Grant no. 2012AA011603. The authors express their sincere thanks.

References

- [1] D. Vavrik, J. Dammer, J. Jakubek et al., "Advanced X-ray radiography and tomography in several engineering applications," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 633, no. 1, pp. S152–S155, 2011.
- [2] G. Bruno, K. Ehrig, H. Haarring et al., "Industrial and synchrotron X-ray CT applications for materials characterisation," in *Proceedings of the Conference on Industrial Computed Tomography*, Wels, Austria, February 2014.
- [3] F. E. Boas and D. Fleischmann, "CT artifacts: causes and reduction techniques," *Imaging in Medicine*, vol. 4, no. 2, pp. 229–240, 2012.

- [4] A. Mouton, N. Megherbi, K. Van Slambrouck, J. Nuyts, and T. P. Breckon, "An experimental survey of metal artefact reduction in computed tomography," *Journal of X-Ray Science and Technology*, vol. 21, no. 2, pp. 193–226, 2013.
- [5] R. Gonzalez and R. Woods, *Digital Image Processing*, Prentice Hall, Upper Saddle River, NJ, USA, 3rd edition, 2007.
- [6] D. Kamal and G. Bindu, "A novel algorithm for image contrast enhancement using histogram equalization," *International Journal of Research in Computer Application & Management*, vol. 1, no. 5, pp. 130–135, 2011.
- [7] S. Das, T. Gulati, and V. Mittal, "Histogram equalization techniques for contrast enhancement: a review," *International Journal of Computer Applications*, vol. 114, no. 10, pp. 32–36, 2015.
- [8] V. Rajamani, P. Babu, and S. Jaiganesh, "A review of various global contrast enhancement techniques for still images using histogram modification framework," *International Journal of Engineering Trends and Technology*, vol. 4, no. 4, pp. 1045–1048, 2013.
- [9] Lalit and A. Gupta, "A review: image enhancement approaches using histogram equalization," *International Journal of Engineering and Computer Science*, vol. 4, no. 8, pp. 14036–14038, 2015.
- [10] S. H. Lim, N. A. Mat Isa, C. H. Ooi, and K. K. V. Toh, "A new histogram equalization method for digital image enhancement and brightness preservation," *Signal, Image & Video Processing*, vol. 9, no. 3, pp. 675–689, 2013.
- [11] J. Lee, S. R. Pant, and H. Lee, "An adaptive histogram equalization based local technique for contrast preserving image enhancement," *The International Journal of Fuzzy Logic & Intelligent Systems*, vol. 15, no. 1, pp. 35–44, 2015.
- [12] M. C. Catalbas, D. Issever, and A. Gulten, "Morphological feature extraction with local histogram equalization," in *Proceedings of the 23rd Signal Processing and Communications Applications Conference (SIU '15)*, pp. 435–438, Malatya, Turkey, May 2015.
- [13] Y.-T. Kim, "Contrast enhancement using brightness preserving bi-histogram equalization," *IEEE Transactions on Consumer Electronics*, vol. 43, no. 1, pp. 1–8, 1997.
- [14] S. C. F. Lin, C. Y. Wong, M. A. Rahman et al., "Image enhancement using the averaging histogram equalization (AVHEQ) approach for contrast improvement and brightness preservation," *Computers & Electrical Engineering*, vol. 46, pp. 356–370, 2014.
- [15] S.-D. Chen and A. R. Ramli, "Contrast enhancement using recursive Mean-Separate histogram equalization for scalable brightness preservation," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1301–1309, 2003.
- [16] M. A. Qadar, Y. Zhaowen, A. Rehman, and M. A. Alvi, "Recursive weighted multi-plateau histogram equalization for image enhancement," *Optik—International Journal for Light and Electron Optics*, vol. 126, no. 24, pp. 5890–5898, 2015.
- [17] K. Singh, R. Kapoor, and S. K. Sinha, "Enhancement of low exposure images via recursive histogram equalization algorithms," *Optik—International Journal for Light and Electron Optics*, vol. 126, no. 20, pp. 2619–2625, 2015.
- [18] Y. Wang, Q. Chen, and B. Zhang, "Image enhancement based on equal area dualistic sub-image histogram equalization method," *IEEE Transactions on Consumer Electronics*, vol. 45, no. 1, pp. 68–75, 1999.
- [19] K. R. Mohan and G. Thirugnanam, "A dualistic sub-image histogram equalization based enhancement and segmentation techniques for medical images," in *Proceedings of the IEEE 2nd International Conference on Image Information Processing (ICIIP '13)*, pp. 566–569, IEEE, Shimla, India, December 2013.
- [20] S.-D. Chen and A. R. Ramli, "Minimum mean brightness error bi-histogram equalization in contrast enhancement," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1310–1319, 2003.
- [21] F. Hossain and M. R. Alsharif, "Minimum mean brightness error dynamic histogram equalization for brightness preserving image contrast enhancement," *International Journal of Innovative Computing, Information & Control*, vol. 5, no. 10, pp. 3263–3274, 2009.
- [22] P.-C. Wu, F.-C. Cheng, and Y.-K. Chen, "A weighting mean-separated sub-histogram equalization for contrast enhancement," in *Proceedings of the International Conference on Biomedical Engineering and Computer Science (ICBECS '10)*, pp. 1–4, IEEE, Wuhan, China, April 2010.
- [23] S. A. Hariprasad, K. S. Arunlal, and V. Ahmed, "Novel weighted mean separated histogram equalization for contrast enhancement of underwater images," *International Journal of Scientific & Engineering Research*, vol. 4, no. 12, pp. 38–42, 2013.
- [24] S. H. Lim, N. A. M. Isa, C. H. Ooi, and K. K. V. Toh, "A new histogram equalization method for digital image enhancement and brightness preservation," *Signal, Image and Video Processing*, vol. 9, no. 3, pp. 675–689, 2015.
- [25] P.-C. Wu, F.-C. Cheng, and Y.-K. Chen, "A weighting mean-separated sub-histogram equalization for contrast enhancement," in *Proceedings of the International Conference on Biomedical Engineering and Computer Science (ICBECS '10)*, pp. 1–4, Wuhan, China, April 2010.
- [26] N. Sengge and H. K. Choi, "Brightness preserving weight clustering histogram equalization," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 3, pp. 1329–1337, 2008.
- [27] S. C. F. Lin, C. Y. Wong, M. A. Rahman et al., "Image enhancement using the averaging histogram equalization (AVHEQ) approach for contrast improvement and brightness preservation," *Computers and Electrical Engineering*, vol. 46, pp. 356–370, 2015.
- [28] A. Abdullah-Al-Wadud, M. H. Kabir, M. A. A. Dewan, and O. Chae, "A dynamic histogram equalization for image contrast enhancement," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 593–600, 2007.
- [29] M. Kim and M. G. Chung, "Recursively separated and weighted histogram equalization for brightness preservation and contrast enhancement," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 3, pp. 1389–1397, 2008.
- [30] Q. Wang and R. K. Ward, "Fast image/video contrast enhancement based on weighted thresholded histogram equalization," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 757–764, 2007.
- [31] G. D. Toderici and J. Yagnik, "Automatic, efficient, temporally-coherent video enhancement for large scale applications," in *Proceedings of the 17th ACM International Conference on Multimedia (MM '09)*, pp. 609–612, ACM, Beijing, China, October 2009.
- [32] S. S. Agaian, K. Panetta, and A. M. Grigoryan, "A new measure of image enhancement," in *Proceedings of the IASTED International Conference on Signal Processing & Communication*, pp. 19–22, 2000.

- [33] S. S. Aghaian, K. Panetta, and A. M. Grigoryan, "Transform-based image enhancement algorithms with performance measure," *IEEE Transactions on Image Processing*, vol. 10, no. 3, pp. 367–382, 2001.
- [34] S. S. Aghaian, B. Silver, and K. A. Panetta, "Transform coefficient histogram-based image enhancement algorithms using contrast entropy," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 741–758, 2007.

Research Article

Neural Architectures for Correlated Noise Removal in Image Processing

Cătălina Cocianu and Alexandru Stan

Computer Science Department, Bucharest University of Economics, 010552 Bucharest, Romania

Correspondence should be addressed to Cătălina Cocianu; ccocianu@ase.ro

Received 21 January 2016; Accepted 24 March 2016

Academic Editor: Marco Perez-Cisneros

Copyright © 2016 C. Cocianu and A. Stan. 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 paper proposes a new method that combines the decorrelation and shrinkage techniques to neural network-based approaches for noise removal purposes. The images are represented as sequences of equal sized blocks, each block being distorted by a stationary statistical correlated noise. Some significant amount of the induced noise in the blocks is removed in a preprocessing step, using a decorrelation method combined with a standard shrinkage-based technique. The preprocessing step provides for each initial image a sequence of blocks that are further compressed at a certain rate, each component of the resulting sequence being supplied as inputs to a feed-forward neural architecture $F_X \rightarrow F_H \rightarrow F_Y$. The local memories of the neurons of the layers F_H and F_Y are generated through a supervised learning process based on the compressed versions of blocks of the same index value supplied as inputs and the compressed versions of them resulting as the mean of their preprocessed versions. Finally, using the standard decompression technique, the sequence of the decompressed blocks is the cleaned representation of the initial image. The performance of the proposed method is evaluated by a long series of tests, the results being very encouraging as compared to similar developments for noise removal purposes.

1. Introduction

There have been proposed a long series of digital image manipulation techniques, general and special tailored ones for different particular purposes. Digital image processing involves procedures including the acquisition and codification of images in digital files and the transmission of the resulting digital files of some communication channels, usually affected by noise [1, 2]. Consequently, a significant part of digital image procedures are devoted to noise removal and image reconstruction, most of them being developed in the framework represented by the assumptions that the superimposed noise is uncorrelated and normally distributed [3, 4]. Our approach is somehow different, keeping the assumption about normality but relaxing the constraint that the superimposed noise affects neighbor image pixels in a correlated way.

There are two basic mathematical characterizations of images, deterministic and statistical. In deterministic image

representation, the image pixels are defined in terms of a certain function, possibly unknown, while, in statistical image representation, the images are specified in probabilistic terms as means, covariances, and higher degree moments [5–7]. In the past years, a series of techniques have been developed in order to involve neural architectures in image compression and denoising processes [8–13].

A neural network is a massively parallel-distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use [14]. The neural networks methodology is of biological inspiration, a neural network resembling the biological brain in two respects; on one hand the knowledge is acquired by the network for its environment through a learning process, and on the other hand the interneuron connection strengths are used to store the acquired knowledge.

The “shrinkage” is a method for reducing the uncorrelated Gaussian noise affecting additively a signal image by soft

thresholding applied to the sparse components [15–17]. Its use in neural network-based approach is intuitively explained by the fact that when only a few of the neurons are simultaneously active, it makes sense to assume that the activities of neurons with small absolute values correspond to noise; therefore they should be set to zero, and only the neurons whose absolute values of their activities are relatively large contain relevant information about the signal.

Recently, a series of correlated noise removal techniques have been reported. Some approaches focus on estimating spatial correlation characteristics of noise for a given image either when noise type and statistics like variance are known [18] or in case the noise variance and spatial spectrum have to be estimated [19] and then use a DCT-based method for noise removal. Wavelet-based approaches mainly include noise prewhitening technique followed by the wavelet-based thresholding [20], additive stationary correlated noise removal by modeling the noise-free coefficients using a multivariate Gaussian Scale Mixture [21], and image denoising using HMM in the wavelet domain based on the concept of signal of interest [22, 23]. Since the sparsity of signals can be exploited for noise removal purpose when different representations are used (Fourier, wavelet, principal components, independent components, etc.), a series of results concerning this property could be of interest in image denoising [24–26] and artifact (noise) removal in magnetic resonance imaging [27].

The outline of the paper is as follows. The general model of image transmission through a noisy corrupted channel is described in Section 2. Each image is transmitted several times as a sequence of equal sized blocks, each block being disturbed by a correlated Gaussian noise whose statistical properties are not known. All variants of each block are submitted to a sequence of transforms that decorrelate, shrink, and average the pixel values.

A special tailored family of feed-forward single-hidden-layer neural networks is described in Section 3, their memories being generated using a supervised learning algorithm of gradient descent type.

A suitable methodology aiming to implement a noise removal method on neural network for image processing purposes is then described in the fourth section of the paper. The proposed methodology was applied to process images from different standard databases, the conclusions experimentally derived from the tests performed on two standard databases, the former containing images of human faces and the latter containing images of landscapes being reported in the next section.

The final section of the paper contains a series of conclusive remarks.

2. Image Preprocessing Based on Decorrelation and Shrinkage Techniques

We assume that the images are transmitted through a noisy channel, each image I being transmitted as a sequence of m d -dimensional blocks, $B_1, B_2, \dots, B_m, I = (B_1, B_2, \dots, B_m)$, and we denote by $\tilde{I} = (\tilde{B}_1, \tilde{B}_2, \dots, \tilde{B}_m)$ the received image.

A working assumption of our model is that the noise modeled by the d -dimensional random vectors $\eta_i, 1 \leq i \leq m$, affects the blocks in a similar way, where $\eta_1, \eta_2, \dots, \eta_m$ are independent identically distributed; $\eta_i \sim N(0, \Sigma), 1 \leq i \leq m$.

In case N images, I^1, I^2, \dots, I^N , are transmitted sequentially through the channel we denote by $(\tilde{I}^i = (\tilde{B}_1^i, \tilde{B}_2^i, \dots, \tilde{B}_m^i), 1 \leq i \leq N)$ the sequence of received disturbed variants. In our model, we adopt the additional working assumption that, for each $1 \leq j \leq m$, $(\tilde{B}_j^1, \tilde{B}_j^2, \dots, \tilde{B}_j^N)$ is a realization of a d -dimensional random vector $\tilde{B}_j^0 = B_j^0 + \eta_j$, where B_j^0 is a random vector of mean μ_j^0 and covariance matrix Σ_j^0 , and that B_j^0 and η_j are independent; therefore the covariance matrix of \tilde{B}_j^0 is $\tilde{\Sigma}_j^0 = \Sigma_j^0 + \Sigma$. The working assumptions included in our model seem to be quite realistic according to the currently used information transmission frameworks. According to the second working assumption, for each index value j , the sequence of blocks $(B_j^1, B_j^2, \dots, B_j^N)$ could represent fragments of possibly different images taken at the counterpart positions, as, for instance, in case of face images the areas of eyes or mouths and so on. Therefore the assumption that each B_j^0 is a random vector corresponds to a model for each particular block, the parameters μ_j^0 and Σ_j^0 expressing the variability existing in the sequence of images I^1, I^2, \dots, I^N at the level of j th block.

On one hand, the maximum likelihood estimates (MLE) of the parameters μ_j^0 and $\tilde{\Sigma}_j^0$ are given by

$$\hat{\mu}_j^0 = \frac{1}{N} \sum_{i=1}^N \tilde{B}_j^i, \quad (1)$$

$$\hat{\Sigma}_j^0 = \frac{1}{N-1} \sum_{i=1}^N (\tilde{B}_j^i - \hat{\mu}_j^0) (\tilde{B}_j^i - \hat{\mu}_j^0)^T, \quad (2)$$

respectively. On the other hand, the values of the parameters μ_j^0 and Σ_j^0 are also unknown and moreover it is quite inconvenient to estimate them before the transmission of the sequence of images is over.

The covariance matrix corresponding to the noise component can be estimated before the transmission is performed by different methods, as, for instance, the white wall method; therefore, without loss of generality, the matrix Σ can be assumed to be known; therefore, $\tilde{\Sigma}_j^0 = \hat{\Sigma}_j^0 - \Sigma$ can be taken as an estimate of Σ_j^0 .

Also, in case each sequence $(\tilde{B}_j^1, \tilde{B}_j^2, \dots, \tilde{B}_j^N)$ is processed separately, we can assume that the data are centered; that is, $\hat{\mu}_j^0 = 0, 1 \leq j \leq m$.

Consequently, the available information in developing a denoising procedure is represented by the sequences $(\tilde{B}_j^1, \tilde{B}_j^2, \dots, \tilde{B}_j^N)$, the estimates $\tilde{\Sigma}_j^0, 1 \leq j \leq m$, and Σ .

In our work we consider the following shrinkage type denoising method.

For each $1 \leq j \leq m$, we denote by A_j a matrix that diagonalizes simultaneously the matrices $\tilde{\Sigma}_j^0$ and Σ .

According to the celebrated W theorem [28, 29], the columns of A_j are eigenvectors of $(\widehat{\Sigma}_j^0)^{-1}\Sigma$ and the following equations hold:

$$(A_j)^T \widehat{\Sigma}_j^0 A_j = I_d, \quad (3)$$

$$(A_j)^T \Sigma A_j = \Lambda^j = \text{diag}(\lambda_1^j, \lambda_2^j, \dots, \lambda_d^j), \quad (4)$$

where $\lambda_1^j, \lambda_2^j, \dots, \lambda_d^j$ are the eigenvalues of the matrix $(\widehat{\Sigma}_j^0)^{-1}\Sigma$. Note that although $(\widehat{\Sigma}_j^0)^{-1}\Sigma$ is not a symmetric matrix, its eigenvalues are proved to be real positive numbers [29].

Let \widetilde{C}_j^i , $1 \leq i \leq N$, be the random vectors:

$$\widetilde{C}_j^i = (A_j)^T \widetilde{B}_j^i = (A_j)^T B_j^i + (A_j)^T \eta_j. \quad (5)$$

Note that the linear transform of matrix $(A_j)^T$ allows obtaining the representation \widetilde{C}_j^i , where the most amount of noise is contained in the second term. Moreover, since

$$\text{Cov}\left((A_j)^T \eta_j, ((A_j)^T \eta_j)^T\right) = (A_j)^T \Sigma A_j = \Lambda^j, \quad (6)$$

the linear transform of matrix $(A_j)^T$ decorrelates the noise components.

Let \widetilde{D}_j^i , $1 \leq i \leq N$, be the sequence of variants of \widetilde{C}_j^i using the code shrinkage method [16], where each entry p , $1 \leq p \leq d$, of \widetilde{D}_j^i is

$$\widetilde{D}_j^i(p) = \text{sgn}\left(\widetilde{C}_j^i(p)\right) \max\left\{0, \left|\widetilde{C}_j^i(p)\right| - \sqrt{2}\lambda_p^j\right\}. \quad (7)$$

Then \widetilde{D}_j^i is a variant of \widetilde{C}_j^i where the noise distributed $N(0, \Lambda^j)$ is partially removed. Since $\widetilde{B}_j^i = ((A_j)^T)^{-1}\widetilde{C}_j^i$ a variant of \widetilde{B}_j^i where the noise was partially removed can be taken as

$$\widehat{B}_j^i = \left((A_j)^T\right)^{-1} \widetilde{D}_j^i. \quad (8)$$

Obviously, from (3) we get $((A_j)^T)^{-1} = \widehat{\Sigma}_j^0 A_j$; that is,

$$\widehat{B}_j^i = \widehat{\Sigma}_j^0 A_j \widetilde{D}_j^i. \quad (9)$$

Note that, although the eigenvalues of $(\widehat{\Sigma}_j^0)^{-1}\Sigma$ are theoretically guaranteed to be positive numbers, in real world applications frequently arise situations when this matrix is ill

conditioned. In order to overpass this difficulty, in our tests we implemented the code shrinkage method using

$$\widetilde{D}_j^i(p) = \begin{cases} \text{sgn}\left(\widetilde{C}_j^i(p)\right) \max\left\{0, \left|\widetilde{C}_j^i(p)\right| - \sqrt{2}\lambda_p^j\right\}, & \lambda_p^j > \varepsilon, \\ \widetilde{C}_j^i(p), & \text{otherwise,} \end{cases} \quad (10)$$

where ε is a conventionally selected positive threshold value. Also, instead of (8) we use

$$\widehat{B}_j^i = \left((A_j)^T\right)^+ \widetilde{D}_j^i, \quad (11)$$

where $((A_j)^T)^+$ is the generalized inverse (Penrose pseudoinverse) of $(A_j)^T$ [30].

In our approach we assumed the source of noise (namely, the communication channel used to transmit the image) can be observed. This hypothesis is frequently used in image restoration techniques [26]. In preprocessing and training stages, undisturbed original versions of the images transmitted are not available; instead, a series of perturbed versions are available and also through white wall technique noise component characteristics may be estimated. Working hypothesis includes the fact that images come from a common probability repartition (maybe a mixture); that is, they share the same statistical characteristics. This hypothesis is frequently used when sets of images are captured and processed [16]. The purpose of this method is, on one hand, to eliminate correlated noise, and, on the other hand, to eliminate the noise from new images transmitted through a communication channel, when they come from the same probability distribution as the images in the initially observed set.

3. Neural Networks Based Approach to Image Denoising

The aim of this section is to present an image denoising method in the framework described in the previous section implemented on a family of standard feed-forward neural architectures $\text{NN}_j : (F_X)_j \rightarrow (F_H)_j \rightarrow (F_Y)_j$, $1 \leq j \leq m$, working in parallel.

Let us assume that $\widetilde{I} = (\widetilde{B}_1, \widetilde{B}_2, \dots, \widetilde{B}_m)$ is the noisy received version of the image $I = (B_1, B_2, \dots, B_m)$ transmitted through the channel. The training process of the architectures NN_j , $1 \leq j \leq m$, is organized such that the resulting memories encode the associations of the type (*input block, sample mean*), the purpose being the noise removal according to the method presented in the previous section.

In order to reduce in some extent the computational complexity, a preprocessing step aiming dimensionality reduction is required. In our work we use L_2 -PCA method to compress the blocks. Since the particular positions of the blocks correspond to different models, their compressed versions could be of different sizes. Indeed, according to (2), the estimates of the autocorrelation matrices $\widehat{\Sigma}_j^0 + \widehat{\mu}_j^0(\widehat{\mu}_j^0)^T$, $1 \leq j \leq m$, are different for different values of the index j ;

therefore, the numbers of the most significant directions are different for different values of index j ; that is, the sizes of the compressed variants of blocks are, in general, different. Consequently, the sizes of $(F_X)_j$ and $(F_Y)_j$ depend on j , these sizes resulting in the preprocessing step by applying the L_2 -PCA method [31, 32].

The hidden neurons influence the error on the nodes to which their output is connected. The use of too many hidden neurons could cause the so-called overfitting effect which means the overestimate of the complexity corresponding to the target problem. Maybe the most unpleasant consequence is that this way the generalization capability is decreased; therefore, the capacity of prediction is degraded too. On the other hand, at least in image processing, the use of fewer hidden neurons implies that less information extracted from the inputs is processed and consequently less accuracy should be expected. Consequently, the determining of the right size of the hidden layer results as a trade-off between accuracy and generalization capacity.

There have been proposed several expressions to compute the number of neurons in the hidden layers [33, 34]. Denoting by $|\cdot|$ the number of elements of the argument, the sizes of the hidden layers $(F_H)_j$ can be computed many ways, some of the most frequent expressions being [34]

$$|(F_H)_j| = 2 \left[\sqrt{(|(F_Y)_j| + 2) |(F_X)_j|} \right], \quad (12a)$$

$$|(F_H)_j| = \left[\sqrt{(|(F_Y)_j| + 2) |(F_X)_j|} + 2 \sqrt{\frac{|(F_X)_j|}{(|(F_Y)_j| + 2)}} \right]. \quad (12b)$$

The aim of the training is that, for each value of the index j to obtain on the output on the layer $(F_Y)_j$, a compressed cleaned version of the input applied to the layer $(F_X)_j$, the output being computed according to the method presented in the previous section.

According to the approach described in the previous section, all blocks of the same index say j are processed by the same compression method yielding to compressed variants, the size of compressed variants being the same for all these blocks. The compressed variants corresponding to the blocks of index j are next fed as inputs to j th neural architecture. Consequently, the denoising process of an image consisting of m blocks is implemented on a family of m neural architectures operating in parallel $(NN_j, 1 \leq j \leq m)$, where $NN_j : (F_X)_j \rightarrow (F_H)_j \rightarrow (F_Y)_j$; the sequence of denoised variants resulted as outputs of the layers $(F_Y)_j$ being next decompressed. The cleaned variant of each input image is taken as the sequence of the decompressed cleaned variants of its blocks.

The preprocessing step producing the compressed variants fed as input blocks is described as follows. For each index value j , the sequence of compressed versions of the blocks $(\tilde{B}_j^1, \tilde{B}_j^2, \dots, \tilde{B}_j^N)$ denoted by $(\widehat{CB}_j^1, \widehat{CB}_j^2, \dots, \widehat{CB}_j^N)$ is

$$\widehat{CB}_j^i = W_j^T \tilde{B}_j^i, \quad i = 1, \dots, N, \quad (13)$$

where the columns of the matrix W_j are the most significant unit eigenvectors of $\tilde{S}_j^0 + \tilde{\mu}_j^0(\tilde{\mu}_j^0)^T$. The most significant unit eigenvectors of $\tilde{S}_j^0 + \tilde{\mu}_j^0(\tilde{\mu}_j^0)^T$ are computed as follows. Let $\theta_1^{(j)} \geq \theta_2^{(j)} \geq \dots \geq \theta_d^{(j)}$ be the eigenvalues of $\tilde{S}_j^0 + \tilde{\mu}_j^0(\tilde{\mu}_j^0)^T$ and $\varepsilon_1 \in (0, 1)$ a conventionally selected threshold value. If t is the smallest value such that (14) holds, then the columns of W_j are unit eigenvectors of $\tilde{S}_j^0 + \tilde{\mu}_j^0(\tilde{\mu}_j^0)^T$ corresponding to the largest t eigenvalues:

$$\frac{1}{\sum_{k=1}^d \theta_k^{(j)}} \sum_{k=t+1}^d \theta_k^{(j)} < \varepsilon_1; \quad (14)$$

therefore, $|(F_X)_j| = t$.

Assuming that the sequence of blocks $(\tilde{B}_j^1, \dots, \tilde{B}_j^N)$ are cleaned versions of $(\tilde{B}_j^1, \tilde{B}_j^2, \dots, \tilde{B}_j^N)$ computed according to (11), we denote by $(\widehat{CB}_j^1, \dots, \widehat{CB}_j^N)$ their compressed variants:

$$\widehat{CB}_j^i = V_j^T \tilde{B}_j^i, \quad i = 1, \dots, N, \quad (15)$$

where the columns of the matrix V_j are the most significant unit eigenvectors of the autocorrelation matrix $(1/N) \sum_{i=1}^N \tilde{B}_j^i (\tilde{B}_j^i)^T$. The most significant eigenvectors of $(1/N) \sum_{i=1}^N \tilde{B}_j^i (\tilde{B}_j^i)^T$ are computed in a similar way as in the compression step applied to input blocks using possibly a different threshold value $\varepsilon_2 \in (0, 1)$. Note that, in tests, the threshold values $\varepsilon_1, \varepsilon_2$ are experimentally tuned to the particular sequence of images.

To summarize, the preprocessing scheme consists of applying L_2 -PCA method to both noisy sequence of blocks $(\tilde{B}_j^1, \tilde{B}_j^2, \dots, \tilde{B}_j^N)$ and their cleaned versions $(\tilde{B}_j^1, \dots, \tilde{B}_j^N)$ causing the sequence of inputs to be applied to the input layer $(F_X)_j$ and to their compressed cleaned versions $(\widehat{CB}_j^1, \dots, \widehat{CB}_j^N)$:

$$\begin{aligned} (\tilde{B}_j^1, \tilde{B}_j^2, \dots, \tilde{B}_j^N) &\xrightarrow{W_j^T} (\widehat{CB}_j^1, \widehat{CB}_j^2, \dots, \widehat{CB}_j^N) \\ &\rightarrow (F_X)_j. \end{aligned} \quad (16)$$

The aim of the training is to produce on each output layer the sequence $(\widehat{CB}_j^1, \dots, \widehat{CB}_j^N)$, the decompressed versions of its blocks being $(V_j \widehat{CB}_j^1, \dots, V_j \widehat{CB}_j^N)$:

$$\begin{aligned} (F_Y)_j &\rightarrow (\widehat{CB}_j^1, \dots, \widehat{CB}_j^N) \\ &\xrightarrow{V_j} (V_j \widehat{CB}_j^1, \dots, V_j \widehat{CB}_j^N); \end{aligned} \quad (17)$$

therefore, the blocks of $(V_j \widehat{CB}_j^1, \dots, V_j \widehat{CB}_j^N)$ are denoised versions of $(\tilde{B}_j^1, \tilde{B}_j^2, \dots, \tilde{B}_j^N)$, respectively.

The training of each neural architecture NN_j is of supervised type using a gradient descent approach, the local memories of $(F_H)_j$ and $(F_V)_j$ being determined using the Levenberg-Marquardt variant of the backpropagation learning algorithm (LM-BP algorithm) [35].

We organized the training process for the m neural networks by transmitting through the channel each available image several times, say p times; the reason of doing that is that this way better estimates of the covariance matrices Σ_j^0 , $1 \leq j \leq m$, of the proposed stochastic models are expected to be obtained.

Consequently, the whole available data is the collection $(\tilde{I}^{i,l} = (\tilde{B}_1^{i,l}, \tilde{B}_2^{i,l}, \dots, \tilde{B}_m^{i,l}), 1 \leq i \leq N, 1 \leq l \leq p)$; therefore, for each index value j , the inputs applied to the j th neural network are the sequence $(\widetilde{CB}_j^{1,1}, \widetilde{CB}_j^{1,2}, \dots, \widetilde{CB}_j^{1,p}, \dots, \widetilde{CB}_j^{N,1}, \dots, \widetilde{CB}_j^{N,p})$ of compressed versions of the blocks $(\tilde{B}_j^{1,1}, \tilde{B}_j^{1,2}, \dots, \tilde{B}_j^{1,p}, \dots, \tilde{B}_j^{N,1}, \dots, \tilde{B}_j^{N,p})$:

$$\widetilde{CB}_j^{i,l} = W_j^T \tilde{B}_j^{i,l}, \quad i = 1, \dots, N, \quad l = 1, \dots, p. \quad (18)$$

The linear compression filter W_j is a matrix whose columns are the most significant unit eigenvectors of the matrix $(1/Np) \sum_{i=1}^N \sum_{l=1}^p \tilde{B}_j^{i,l} (\tilde{B}_j^{i,l})^T$.

Let $(\hat{B}_j^{1,1}, \dots, \hat{B}_j^{1,p}, \dots, \hat{B}_j^{N,1}, \dots, \hat{B}_j^{N,p})$ be the sequence of the cleaned variants of $(\tilde{B}_j^{1,1}, \tilde{B}_j^{1,2}, \dots, \tilde{B}_j^{1,p}, \dots, \tilde{B}_j^{N,1}, \dots, \tilde{B}_j^{N,p})$ computed using (11) and, for each $1 \leq i \leq N$, let M_j^i be the sample mean of cleaned blocks $(\hat{B}_j^{i,1}, \dots, \hat{B}_j^{i,p})$:

$$M_j^i = \frac{1}{p} \sum_{l=1}^p \hat{B}_j^{i,l}. \quad (19)$$

We denote by V_j a linear compression filter whose columns are the most significant unit eigenvectors of the matrix $(1/N) \sum_{i=1}^N M_j^i (M_j^i)^T$ computed in a similar way as (15) using a threshold value $\varepsilon_2 \in (0, 1)$ and let $CM_j^i = V_j^T M_j^i$.

The learning process for each neural architecture NN_j , $1 \leq j \leq m$, is developed to encode the associations $(\widetilde{CB}_j^{k,1}, \dots, \widetilde{CB}_j^{k,p}) \rightarrow CM_j^k$, $1 \leq k \leq N$. The reason of using the means M_j^k , $1 \leq k \leq N$, and their corresponding compressed versions instead of the associations $(\widetilde{CB}_j^{k,1}, \dots, \widetilde{CB}_j^{k,p}) \rightarrow (\widetilde{CB}_j^{k,1}, \dots, \widetilde{CB}_j^{k,p})$, $1 \leq k \leq N$, resides in the fact that taking the means and their compressed versions some amount of noise is expected to be removed, for each value of the index j ; that is, the compressed versions of the means are expected to be better cleaned variants of the compressed blocks.

Summarizing, the memory of each neural architecture NN_j is computed by the Levenberg-Marquardt algorithm applied to the input/output sequence $\{(\widetilde{CB}_j^{1,1}, CM_j^1), \dots, (\widetilde{CB}_j^{1,p}, CM_j^1), \dots, (\widetilde{CB}_j^{N,1}, CM_j^N), \dots, (\widetilde{CB}_j^{N,p}, CM_j^N)\}$, $1 \leq j \leq m$.

Once the training phase is over, the family of NN_j 's is used to remove the noise from a noisy version of an image $\tilde{I} = (\tilde{B}_1, \tilde{B}_2, \dots, \tilde{B}_m)$ received through the channel according to the following scheme. Let $I = (B_1, B_2, \dots, B_m)$ be the initial image transmitted through the channel and $\tilde{I} = (\tilde{B}_1, \tilde{B}_2, \dots, \tilde{B}_m)$ the received noisy version.

Step 1. Compress each block \tilde{B}_j of \tilde{I} using the filter W_j and get its compressed version \widetilde{CB}_j ; that is, $(\widetilde{CB}_1, \widetilde{CB}_2, \dots, \widetilde{CB}_m)$ is a dynamically block-compressed version of \tilde{I} .

Step 2. Apply $(\widetilde{CB}_1, \widetilde{CB}_2, \dots, \widetilde{CB}_m)$ as inputs to the architectures NN_j 's, \widetilde{CB}_j applied as input to the layer $(F_X)_j$, $1 \leq j \leq m$, and get the outputs RB_j 's.

Step 3. Decompress each block RB_j using the decompression filter V_j , $1 \leq j \leq m$.

Step 4. Get $\hat{I} = (V_1 \cdot RB_1, \dots, V_m \cdot RB_m)$ the cleaned version of \tilde{I} .

4. Description of the Methodology Applied in the Implementations of the Proposed Method on Neural Architectures

The aim of this section is to describe the methodology followed in implementing the neural network-based noise removal method for image processing purposes. The proposed methodology was applied to process images from different standard databases, the conclusions experimentally derived from the tests performed on two standard databases, the former containing images of human faces and the latter containing images of landscapes being reported in the next section.

We performed the experiments according to the following methodology.

(1) The quality of the a certain test image $T = (t(x, y))$ versus a reference image $R = (r(x, y))$ of the same size (n_x, n_y) is evaluated in terms of the Signal-to-Noise Ratio (SNR), Peak Signal-to-Noise Ratio (PSNR), Root Mean Squared Signal-to-Noise Ratio (SNR_RMS) indicators [36], and the Structural Similarity Metric (SSIM) [37], where

$$\begin{aligned} \text{SNR}(R, T) &= 10 * \log_{10} \left[\frac{\sum_{x=1}^{n_x} \sum_{y=1}^{n_y} (r(x, y))^2}{\sum_{x=1}^{n_x} \sum_{y=1}^{n_y} (r(x, y) - t(x, y))^2} \right], \\ \text{PSNR}(R, T) &= 10 \\ &* \log_{10} \left[\frac{\max(r(x, y))^2}{(1/(n_x * n_y)) \sum_{x=1}^{n_x} \sum_{y=1}^{n_y} (r(x, y) - t(x, y))^2} \right], \quad (20) \\ \text{SNR_RMS}(R, T) &= \sqrt{\frac{\sum_{x=1}^{n_x} \sum_{y=1}^{n_y} (r(x, y))^2}{\sum_{x=1}^{n_x} \sum_{y=1}^{n_y} (r(x, y) - t(x, y))^2}}. \end{aligned}$$

Let x and y be spatial patches extracted from the images R and T , respectively. The two patches correspond to the same

spatial window of the images R and T . The original standard SSIM value computed for the patches x and y is defined by

$$\text{SSIM}(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \cdot \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \quad (21)$$

where μ_x denotes the mean value of x , σ_x is the standard deviation of x , and σ_{xy} represents the cross-correlation of the mean shifted patches $x - \mu_x$ and $y - \mu_y$. The constants C_1 and C_2 are small positive numbers included to avoid instability when either $\mu_x^2 + \mu_y^2$ or $\sigma_x^2 + \sigma_y^2$ is very close to zero, respectively. The overall SSIM index for the images R and T is computed as the mean value of the SSIM measures computed for all pairs of patches x and y of R and T , respectively.

(2) The size of the blocks and the model of noise in transmitting data through the channel are selected for each database. The size of the blocks is established by taking into account the size of the available images in order to assure reasonable complexity to the noise removal process. In our tests the size of input blocks is about 150 and the sizes of images are 135×100 in case of the database containing images of human faces and 154×154 in case of the database containing images of landscapes. We assumed that the components of the noise η induced by the channel are possibly correlated; in our tests, the noise model is of Gaussian type, $\eta \sim N(0, \Sigma)$, where Σ is a symmetric positive defined matrix.

(3) The compression thresholds $\varepsilon_1, \varepsilon_2$ in (14) and (15) are established in order to assure some desired accuracy. In our tests we used $\varepsilon_1 = c_1 * 10^{-4}$, $\varepsilon_2 = c_2 * 10^{-4}$, where c_1, c_2 are positive constants. The reason for selecting different magnitude orders of these thresholds stems from the fact that ε_1 is used in compressing noise affected images, while ε_2 is used for compressing noise cleaned images [32]. The sizes of the input and output layers $(F_X)_j, (F_Y)_j$ of the neural network NN_j result in terms of the established values of ε_1 and ε_2 accordingly.

(4) The quality evaluation of the preprocessing step consisting in noise cleaning data is performed in terms of the indicators (20) and (21), by comparing the initial data $I = (B_1, B_2, \dots, B_m)$ against the noisy transmitted images $\tilde{I} = (\tilde{B}_1, \tilde{B}_2, \dots, \tilde{B}_m)$ through the channel and $I = (B_1, B_2, \dots, B_m)$ against their corresponding cleaned versions $\hat{I} = (V_1 \cdot RB_1, \dots, V_m \cdot RB_m)$, respectively.

(5) In order to implement the noise removal method on a family of neural networks $\text{NN}_j : (F_X)_j \rightarrow (F_H)_j \rightarrow (F_Y)_j$, $1 \leq j \leq m$, the sizes of the input $(F_X)_j$ and the output layers $(F_Y)_j$ are determined by L_2 -PCA compression/decompression method and the established values of $\varepsilon_1, \varepsilon_2$. The sizes of the layers $(F_H)_j$ are determined as approximations of the recommended values cited in the published literature (12a) and (12b). In order to assure a reasonable tractability of the data, in our tests we were forced to use a less number of neurons than it is recommended, on the hidden layers $(F_H)_j$.

For fixed values of $\varepsilon_1, \varepsilon_2$, the use of the recommended number of neurons as in (12a) and (12b) usually yields to

either the impossibility of implementing the learning process or to too lengthy training processes. Therefore, in such case we are forced to reconsider the values of $\varepsilon_1, \varepsilon_2$ by increasing them, therefore decreasing the numbers of neurons on the input and the output layers and consequently the number of neurons on the hidden layers too. Obviously, by reconsidering this way the values of $\varepsilon_1, \varepsilon_2$, inherently imply that some larger amount of information about data is lost. The effects of losing information are manifold, one of them being that the cleaned versions resulted from decompressing the outputs of NN_j 's yield to poorer approximation \hat{I} of the initial image I .

This way we arrive at the conclusion that, in practice, we have to solve a trade-off problem between the magnitude of the compression rates and the number of neurons on the hidden layers $(F_H)_j$'s. In order to solve this trade-off, in our tests we used smaller numbers of neurons than recommended on the hidden layers and developed a comparative analysis on the quality of the resulting cleaned images.

(6) The activation functions of the neurons belonging to the hidden and output layers can be selected from very large family. In our tests, we considered the logistic type to model the activation functions of the neurons belonging to the hidden layers and the unit functions to model the outputs of the neurons belonging to the output layers. Also, the learning process involved the task of splitting the available data into training, validation, and test data. In our tests the sizes of the subcollections were 80%, 10%, and 10%, respectively.

(7) The evaluation of the overall quality of the noise removal process implemented on the set of neural networks, as previously described, is performed in terms of the indicators (20) and (21), on one hand by comparing the initial data $I = (B_1, B_2, \dots, B_m)$ to the noisy transmitted images $\tilde{I} = (\tilde{B}_1, \tilde{B}_2, \dots, \tilde{B}_m)$ through the channel and on the other hand by comparing $I = (B_1, B_2, \dots, B_m)$ to their cleaned versions $\hat{I} = (V_1 \cdot RB_1, \dots, V_m \cdot RB_m)$.

(8) The comparative analysis between the performances corresponding to the decorrelation and shrinkage method and its implementation on neural networks is developed in terms of the indicators (20) and (21).

5. Experimentally Derived Conclusions on the Performance of the Proposed Method

In this section we present the results in evaluating both the quality of the proposed decorrelation and shrinkage method and the power of the neural network-based approach in simulating it for noise removal purposes. The tests were performed in a similar way on two standard databases, the former, referred to as Senthil, containing images of 5 human faces and 16 images for each person [38] and the latter containing 42 images of landscapes [39]. In case of the Senthil database, the preprocessing step used 75 images; for each human face 15 of its available versions are being used. The tests performed in order to evaluate the quality of the trained family of neural networks used the rest of 5 images, one for each person. In case of the database containing images of landscapes, we identified three types of quite similar images,

TABLE 1

| The maximum number of epochs | The minimum value of the performance (Jacobian computation) | The maximum validation failures | The minimum performance gradient | The initial/maximum μ factor (in the LM adaptive learning rule) |
|------------------------------|---|---------------------------------|----------------------------------|---|
| 1000 | 0 | 5 | 10^{-5} | $10^{-3}/10^{10}$ |

and we used 13 images of each type in the training process, the tests being performed on the rest of three ones.

The sizes of hidden layers were set to smaller values than recommended by (12a) and (12b). For instance, when $\varepsilon_1 \approx 10^{-4}$ and $\varepsilon_2 \approx 10^{-4}$ the resulting sizes of the layers $|(F_X)_j|$ and $|(F_Y)_j|$ are about 115 and 30, respectively, the recommended sizes of the layers $|(F_H)_j|$ being about 65.

The results of a long series of tests pointed out that one can use hidden layers of smaller sizes than recommended without decreasing dramatically the accuracy. For instance, in this work, we used only half of recommended sizes; that is,

$$|(F_H)_j| = \left[\frac{\sqrt{(|(F_Y)_j| + 2)|(|F_X)_j|)} + 2\sqrt{|(F_X)_j| / (|(F_Y)_j| + 2)}}{2} \right]. \quad (22)$$

In our test, the memory of each neural architecture is computed by the LM-BP algorithm, often the fastest variant of the backpropagation algorithm and one of the most commonly used in supervised learning. The available data was split into training set, validation set, and test set, the sizes of the subcollections being 80%, 10%, and 10%, respectively. The main parameters of the LM-BP training process are specified in Table 1.

In order to experimentally establish the quality of the proposed method, a comparative analysis against three of the most used and suitable algorithms for correlated noise removal, namely, BM3D (block-matching and 3D filtering [25]), NLMF (Nonlocal Means Noise Filtering [40]), and ProbShrink (correlated noise removal algorithm using nondecimated wavelet transform and generalized Laplacian [22]), was conducted. The reported results include both quantitative and qualitative comparisons.

In the following, we summarize some of our results.

(a) The quality evaluation of the preprocessing step in terms of the indicators (20) and (21) is as follows:

(1) In Figure 1(a), a sample of five face images belonging to the Senthil database is presented, their cleaned versions resulted from applying the decorrelation and shrinkage method being shown in Figure 1(e), where each image was transmitted 30 times through the channel. In Figures 1(b), 1(c), and 1(d) the restored versions resulting from applying the NLMF algorithm, ProbShrink algorithm, and BM3D method, respectively, are depicted. Table 2 contains the

values of the indicators (20) and (21) corresponding to these five pairs of noisy-cleaned versions of these images.

Note that, on average, the best results were obtained when our method was used.

(2) A sample of three images of landscapes, one from each class, is presented in Figure 2(a) together with their cleaned versions resulting from applying the decorrelation and shrinkage method shown in Figure 2(e), where each image was transmitted 30 times through the channel. In Figure 2(b) the restored variants using NLMF algorithm are exhibited, while in Figure 2(c) the restored variants using ProbShrink method are shown. The cleaned version using the BM3D algorithm is presented in Figure 2(d). The values of the indicators (20) and (21) are given in Table 3.

Note that, on average, the best results were obtained when our method was used.

(b) As it was previously described, the images resulting from the preprocessing step are used in the supervised training of the family of neural networks. Once the training process is over, the family of neural networks are used to remove noise from new unseen-test images. Obviously, it is impossible to guarantee that the new test images share the same statistical properties with the images used during the training process, the unique criterion being that they are visually enough similar. In order to take into account this constraint, we split each of these two databases containing similar images into training and testing subsets, the sizes being 75/5 for Senthil dataset and 39/3 for the second database.

(1) The test images from the Senthil database and their versions resulting from applying the preprocessing step are shown in Figures 3(a) and 3(b), respectively. Their cleaned versions computed by the resulting family of trained neural networks are shown in Figure 3(f), while their restored versions when NLMF algorithm, ProbShrink method, and BM3D method are used are presented in Figures 3(c), 3(d), and 3(e), respectively. In terms of the indicators (20) and (21), the results are summarized in Table 4. Note that in this case our method, ProbShrink algorithm, and BM3D method produce similar results, according to both SNR measure and SSIM metric.



FIGURE 1

(2) Similar tests were performed on the database containing images of landscapes. The tests were performed on three new images shown in Figure 4(a), the results of the preprocessing step being given in Figure 4(b). The cleaned versions computed by the resulting family of trained neural networks are shown in Figure 4(f) and the clean versions given by the BM3D algorithm are presented in Figure 4(e). The results obtained when the NLMF algorithm and ProbShrink method are used are displayed in Figures 4(c)

and 4(d), respectively. The numerical evaluation in terms of the indicators (20) and (21) is summarized in Table 5.

Note that, in this case, the BM3D algorithm proved to smooth the results too much. Also, the images obtained when NLMF algorithm was used are of poor visual quality. The ProbShrink algorithm performed better than BM3D and NLMF, but, on average, the best results were obtained when our method was used.

TABLE 2

| | SNR-RMS (the mean value) | SNR (the mean value) | Mean Peak SNR (the mean value) | SSIM (the mean value) |
|---|--------------------------|----------------------|--------------------------------|-----------------------|
| Noisy images versus original images | 7.3128 | 17.2193 | 26.3217 | 0.5838 |
| Cleaned images versus original images (NLMF) | 12.1367 | 21.6369 | 30.8944 | 0.8377 |
| Cleaned images versus original images (ProbShrink) | 15.3787 | 23.7071 | 33.1043 | 0.8782 |
| Cleaned images versus original images (BM3D) | 15.0573 | 23.5242 | 33.3160 | 0.8867 |
| Cleaned images versus original images (the proposed method) | 19.5967 | 25.8067 | 35.0112 | 0.9163 |

TABLE 3

| | SNR-RMS (the mean value) | SNR (the mean value) | Mean Peak SNR (the mean value) | SSIM (the mean value) |
|---|--------------------------|----------------------|--------------------------------|-----------------------|
| Noisy images versus original images | 7.1013 | 17.0083 | 24.6427 | 0.6868 |
| Cleaned images versus original images (NLMF) | 7.5050 | 17.4992 | 25.8548 | 0.6503 |
| Cleaned images versus original images (ProbShrink) | 9.6534 | 19.7858 | 27.7324 | 0.8002 |
| Cleaned images versus original images (BM3D) | 8.7304 | 18.8084 | 27.6967 | 0.7389 |
| Cleaned images versus original images (the proposed method) | 20.8267 | 26.2779 | 34.0215 | 0.9341 |

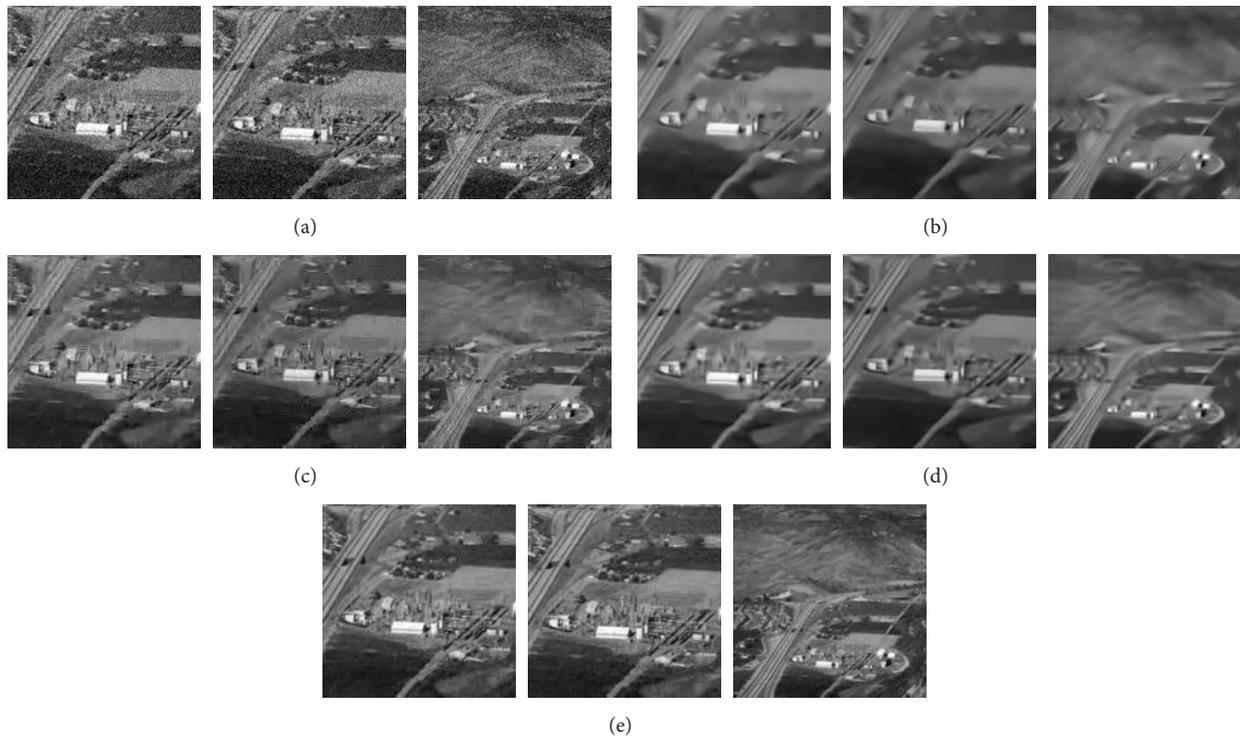


FIGURE 2



(a)



(b)



(c)



(d)



(e)



(f)

FIGURE 3

TABLE 4

| | SNR-RMS/new image | SNR/new image | Mean Peak SNR/new image | SSIM/new image |
|--|-------------------|---------------|-------------------------|----------------|
| Noisy images versus original images | 7.1352 | 17.0681 | 26.3356 | 0.5700 |
| | 7.7979 | 17.8395 | 26.4804 | 0.5693 |
| | 6.7452 | 16.5800 | 26.2165 | 0.5733 |
| | 9.0387 | 19.1221 | 26.4122 | 0.6410 |
| | 6.1524 | 15.7809 | 26.5486 | 0.5513 |
| Cleaned images (using the preprocessing step) versus original images | 8.4056 | 18.4914 | 27.8057 | 0.6328 |
| | 9.6450 | 19.6860 | 28.3873 | 0.6411 |
| | 8.3065 | 18.3883 | 28.0880 | 0.6392 |
| | 9.5696 | 19.6179 | 26.9517 | 0.6643 |
| | 7.4780 | 17.4758 | 28.2541 | 0.6308 |
| Cleaned images versus original images (NLMF) | 12.1067 | 21.6605 | 31.0955 | 0.8474 |
| | 13.6918 | 22.7292 | 31.5051 | 0.8448 |
| | 11.9228 | 21.5275 | 31.3547 | 0.8424 |
| | 14.5670 | 23.2674 | 30.6666 | 0.8588 |
| | 10.8248 | 20.6884 | 31.6597 | 0.8619 |
| Cleaned images versus original images (ProbShrink) | 15.6521 | 23.8914 | 33.2490 | 0.8834 |
| | 15.5453 | 23.8320 | 32.5356 | 0.8648 |
| | 14.6064 | 23.2909 | 33.0326 | 0.8705 |
| | 17.5242 | 24.8728 | 32.2178 | 0.8836 |
| | 13.2653 | 22.1486 | 33.5365 | 0.8812 |
| Cleaned images versus original images (BM3D) | 14.8333 | 23.4248 | 33.34 | 0.8824 |
| | 16.3388 | 24.2644 | 33.59 | 0.8854 |
| | 14.7108 | 23.3527 | 33.68 | 0.8991 |
| | 17.7965 | 25.0067 | 33.00 | 0.8525 |
| | 13.4046 | 22.4632 | 33.63 | 0.9074 |
| Cleaned images (using NN's) versus original images (the proposed method) | 13.7982 | 22.7964 | 32.2178 | 0.8850 |
| | 16.9576 | 24.5873 | 33.0786 | 0.8781 |
| | 16.5712 | 24.3871 | 33.8409 | 0.8968 |
| | 12.6325 | 22.0298 | 29.6259 | 0.8454 |
| | 13.4488 | 22.5737 | 33.6427 | 0.8939 |

TABLE 5

| | SNR-RMS/new image | SNR/new image | Mean Peak SNR/new image | SSIM/new image |
|--|-------------------|---------------|-------------------------|----------------|
| Noisy images versus original images | 6.7603 | 16.5994 | 24.6606 | 0.6564 |
| | 6.5259 | 16.2928 | 24.6103 | 0.6796 |
| | 7.7468 | 17.7824 | 24.7497 | 0.6157 |
| Cleaned images versus original images | 6.8194 | 16.6749 | 24.7947 | 0.6590 |
| | 7.8383 | 17.8845 | 26.2810 | 0.7203 |
| | 7.3444 | 17.3191 | 24.3228 | 0.6990 |
| Cleaned images versus original images (NLMF) | 7.7084 | 17.7393 | 26.0425 | 0.6773 |
| | 6.9041 | 16.7822 | 25.3903 | 0.6025 |
| | 8.2267 | 18.3045 | 25.4729 | 0.6830 |
| Cleaned images versus original images (ProbShrink) | 9.5398 | 19.6813 | 27.7907 | 0.8060 |
| | 9.0598 | 19.1423 | 27.6171 | 0.7732 |
| | 10.0007 | 20.0039 | 27.5228 | 0.8218 |
| Cleaned images versus original images (BM3D) | 9.0589 | 19.1415 | 27.95 | 0.7265 |
| | 7.8517 | 17.8993 | 26.99 | 0.6969 |
| | 9.7064 | 19.7412 | 27.89 | 0.7688 |
| Cleaned images (using NN's) versus original images (the proposed method) | 9.4599 | 19.5177 | 27.6939 | 0.8090 |
| | 13.1507 | 22.3790 | 30.8164 | 0.8770 |
| | 10.0606 | 20.0525 | 27.1269 | 0.8412 |

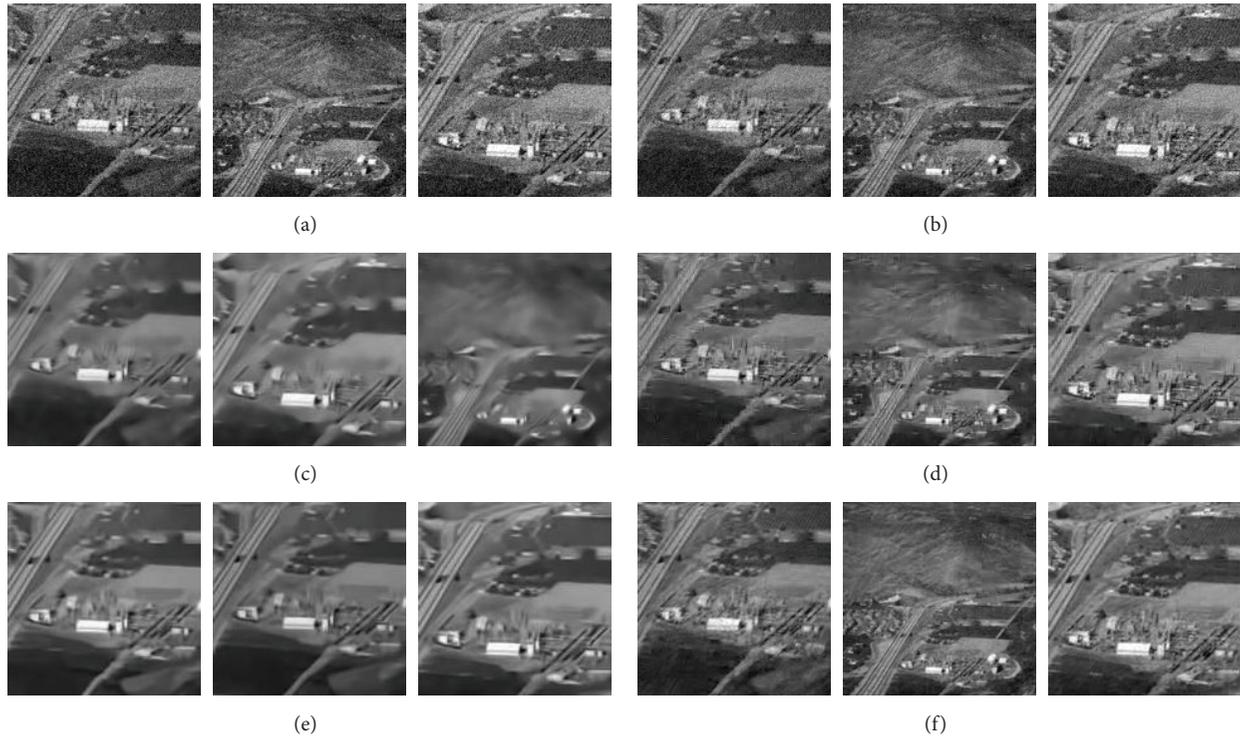


FIGURE 4

6. Conclusive Remarks and Suggestions for Further Work

The proposed method combines the decorrelation and shrinkage techniques to neural network-based approaches for noise removal purposes. The images are assumed to be transmitted as sequences of blocks of equal sizes, each block being distorted by a stationary statistical correlated noise and some amount of the noise being partially removed using the method that combines noise decorrelation and standard shrinkage technique. The preprocessing step provides, for each initial image, a sequence of blocks that are further PCA-compressed at a certain rate, each component of the resulting sequence being supplied as inputs to a feed-forward neural architecture $F_X \rightarrow F_H \rightarrow F_Y$. Therefore, each indexed block is processed by a neural network corresponding to that index value. The local memories of the neurons of the layers F_H and F_Y are generated through a supervised learning process based on the compressed versions of blocks of the same index value supplied as inputs and the compressed versions of them resulting as the mean of their preprocessed versions. Finally, using the standard PCA-decompression technique, the sequence of the decompressed blocks is the cleaned representation of the initial image. The performance of the proposed method is evaluated by a long series of tests, the results being very encouraging as compared to similar developments for noise removal purposes. The evaluation of the amount of the noise removed is done in terms of some of the most frequently used similarity indicators, SNR, SNR-RMS, Peak SNR, and SSIM.

The results produced by applying the proposed method were compared to those produced by applying three of the most widely used algorithms for eliminating correlated noise. NLMF algorithm consistently produces weaker results than the proposed method. Using ProbShrink or BM3D, the results are similar to or weaker than those yielded by the proposed method, in both quality and quantity.

The long series of tests proved good results of the above-described methodology entailing the hope that further and possibly more sophisticated extensions can be expected to improve it. Among several possible extensions, some work is still in progress concerning the use of different output functions for the hidden and output neurons and the use of more hidden layers in the neural architectures. Also, some other compression techniques combined with new techniques for feature extraction as well as the use of other learning schemes to generate the local memories of the neurons are expected to allow the removal of a larger amount of noise.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

A major contribution to the research work reported in this paper belongs to Mrs. Luminita State, a Professor and a Ph.D. Distinguished Member of the Romanian academic

community; Professor Luminita State passed away in January 2016. The authors will always remember her amazing spirit, as well as her brilliant mind. May God rest her soul in peace!

References

- [1] W. Pratt, *Digital Image Processing*, Wiley-Interscience, Hoboken, NJ, USA, 4th edition, 2007.
- [2] E. López-Rubio, "Restoration of images corrupted by Gaussian and uniform impulsive noise," *Pattern Recognition*, vol. 43, no. 5, pp. 1835–1846, 2010.
- [3] L. State, C. Cocianu, C. Săraru, and P. Vlamos, "New approaches in image compression and noise removal," in *Proceedings of the 1st International Conference on Advances in Satellite and Space Communications (SPACOMM '09)*, pp. 96–101, IEEE, Colmar, France, July 2009.
- [4] Z. H. Shamsi and D.-G. Kim, "Multiscale hybrid nonlocal means filtering using modified similarity measure," *Mathematical Problems in Engineering*, vol. 2015, Article ID 318341, 17 pages, 2015.
- [5] P. Fieguth, *Statistical Image Processing and Multidimensional Modeling*, Springer, New York, NY, USA, 2011.
- [6] L. Tan and J. Jiang, *Digital Signal Processing. Fundamentals and Applications*, Academic Press, Elsevier, 2nd edition, 2013.
- [7] S. Kay, *Fundamentals of Statistical Signal Processing, Volume III: Practical Algorithm Development*, Prentice Hall, New York, NY, USA, 2013.
- [8] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural networks—a review," *Pattern Recognition*, vol. 35, no. 10, pp. 2279–2301, 2002.
- [9] F. Hussain and J. Jeong, "Efficient deep neural network for digital image compression employing rectified linear neurons," *Journal of Sensors*, vol. 2016, Article ID 3184840, 7 pages, 2016.
- [10] A. J. Hussain, D. Al-Jumeily, N. Radi, and P. Lisboa, "Hybrid neural network predictive-wavelet image compression system," *Neurocomputing*, vol. 151, no. 3, pp. 975–984, 2015.
- [11] S. Bhattacharyya, P. Pal, and S. Bhowmick, "Binary image denoising using a quantum multilayer self organizing neural network," *Applied Soft Computing Journal*, vol. 24, pp. 717–729, 2014.
- [12] Y. Li, J. Lu, L. Wang, and Y. Takashi, "Denoising by using multineural networks for medical X-ray imaging applications," *Neurocomputing*, vol. 72, no. 13–15, pp. 2884–2891, 2009.
- [13] I. Turkmen, "The ANN based detector to remove random-valued impulse noise in images," *Journal of Visual Communication and Image Representation*, vol. 34, pp. 28–36, 2016.
- [14] S. Haykin, *Neural Networks A Comprehensive Foundation*, Prentice Hall, 1999.
- [15] Y. Wu, B. H. Tracey, P. Natarajan, and J. P. Noonan, "Fast block-wise SURE shrinkage for image denoising," *Signal Processing*, vol. 103, pp. 45–59, 2014.
- [16] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, New York, NY, USA, 2001.
- [17] L. Shang, D.-S. Huang, C.-H. Zheng, and Z.-L. Sun, "Noise removal using a novel non-negative sparse coding shrinkage technique," *Neurocomputing*, vol. 69, no. 7–9, pp. 874–877, 2006.
- [18] N. N. Popomarenko, V. V. Lukin, A. A. Zelensky, J. T. Astola, and J. T. Astola, "Adaptive DCT-based filtering of images corrupted by spatially correlated noise," in *Image Processing: Algorithms and Systems VI*, vol. 6812 of *Proceedings of SPIE*, San Jose, Calif, USA, January 2008.
- [19] N. N. Popomarenko, V. V. Lukin, K. O. Egiazarian, and J. T. Astola, "A method for blind estimation of spatially correlated noise characteristics," in *Image Processing: Algorithms and Systems VIII*, vol. 7532 of *Proceedings of SPIE*, February 2010.
- [20] I. M. Johnstone and B. W. Silverman, "Wavelet threshold estimators for data with correlated noise," *Journal of the Royal Statistical Society. Series B. Methodological*, vol. 59, no. 2, pp. 319–351, 1997.
- [21] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1338–1351, 2003.
- [22] A. Pižurica and W. Philips, "Estimating the probability of the presence of a signal of interest in multiresolution single- and multiband image denoising," *IEEE Transactions on Image Processing*, vol. 15, no. 3, pp. 654–665, 2006.
- [23] B. Goossens, Q. Luong, A. Pizurica, and W. Philips, "An improved non-local denoising algorithm," in *Proceedings of the International Workshop on Local and Non-Local Approximation in Image Processing (LNLA '08)*, pp. 143–156, Lausanne, Switzerland, August 2008.
- [24] M. Jansen, *Noise Reduction by Wavelet Thresholding*, Springer, Berlin, Germany, 2001.
- [25] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [26] M. P. S. Chawla, "PCA and ICA processing methods for removal of artifacts and noise in electrocardiograms: a survey and comparison," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 2216–2226, 2011.
- [27] L. Griffanti, G. Salimi-Khorshidi, C. F. Beckmann et al., "ICA-based artefact removal and accelerated fMRI acquisition for improved resting state network imaging," *NeuroImage*, vol. 95, pp. 232–247, 2014.
- [28] P. Common and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, Academic Press, Elsevier, 2010.
- [29] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, 2nd edition, 1990.
- [30] J. E. Gentle, *Matrix Algebra. Theory, Computations, and Applications in Statistics*, Springer Texts in Statistics, Springer, New York, NY, USA, 2007.
- [31] I. T. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics, Springer, Berlin, Germany, 2nd edition, 2002.
- [32] C. Cocianu, L. State, and P. Vlamos, "Neural implementation of a class of PCA learning algorithms," *Economic Computation and Economic Cybernetics Studies and Research* no. 3/2009, 2009.
- [33] K. Gnana Sheela and S. N. Deepa, "Review on methods to fix number of hidden neurons in neural networks," *Mathematical Problems in Engineering*, vol. 2013, Article ID 425740, 11 pages, 2013.
- [34] G.-B. Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 274–281, 2003.
- [35] G. A. F. Seber and C. J. Wild, *Nonlinear Regression*, John Wiley & Sons, New York, NY, USA, 2003.
- [36] R. Gonzales and R. Woods, *Digital Image Processing*, Prentice Hall, 5th edition, 2008.

- [37] 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.
- [38] <http://www.geocities.ws/senthilirtt/Senthil%20Face%20Database%20Version1>.
- [39] <http://sipi.usc.edu/database/database.php?volume=sequences>.
- [40] 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. 2, pp. 60–65, IEEE, San Diego, Calif, USA, June 2005.

Research Article

Solving Adaptive Image Restoration Problems via a Modified Projection Algorithm

Hao Yang,^{1,2} Xueping Luo,³ and Leiting Chen^{1,2}

¹*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China*

²*Sichuan Provincial Key Laboratory of Digital Media, Chengdu, Sichuan 611731, China*

³*College of Computer Science and Technology, Southwest University for Nationalities, Chengdu, Sichuan 610041, China*

Correspondence should be addressed to Hao Yang; vhaoyang@gmail.com

Received 26 January 2016; Accepted 9 March 2016

Academic Editor: Daniel Zaldivar

Copyright © 2016 Hao Yang 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 introduce a new general TV regularizer, namely, generalized TV regularization, to study image denoising and nonblind image deblurring problems. In order to discuss the generalized TV image restoration with solution-driven adaptivity, we consider the existence and uniqueness of the solution for mixed quasi-variational inequality. Moreover, the convergence of a modified projection algorithm for solving mixed quasi-variational inequalities is also shown. The corresponding experimental results support our theoretical findings.

1. Introduction

Digital image restoration plays an important role in many applications of sciences and engineering such as medical and astronomical imaging, film restoration, and image and video coding. Recovering an image from a degraded image is usually an ill-posed inverse problem and it should be dealt with through selecting a suitable regularizer. Since the work of Rudin, Osher, and Fatemi (ROF) in [1], the regularization methods based on total variation (TV) have known a success, mostly due to their ability to preserve edges in the image. In recent years, a number of research works have been proposed in the field of TV regularization approaches, which are used for the task of image denoising and nonblind image deblurring. Aujol et al. [2] replaced L_2 norm of the data fidelity term by L_1 norm to modify the ROF functional model. The TV regularization approaches in [3, 4] can be described by means of locally dependent constraint sets; that is, the functional is adaptive to the input data. Another class of approaches are the nonlocal methods [5, 6] including nonlocal variants of TV regularization. [7–9] extended TV regularization to second- or higher-order cases. These works

mentioned above considered TV regularization approaches for solving the image denoising problems. In addition, these approaches can also be utilized for the image deblurring; see, for example, [10–12]. Chambolle [10] proposed algorithm for minimizing the TV model and applied the algorithm to image zooming. A TV deblurring approach with adaptive choice of the regularization parameter was presented in [12]. In all these literatures, the image restoration problem is always regarded as optimization problem using discrete TV regularization. How to solve such optimization problem with a TV regularization, which is fundamental and crucial, is the core problem in our discussion.

It is well known that the theory of variational inequality has been developed as a class of important tools for the study of minimization problems; see, for example, [13]. Among such variational inequalities, inverse variational inequalities, mixed variational inequalities, and quasi-variational inequalities are very significant generalizations, which have been applied to a wide range of problems, such as mechanics, economics, finance, optimal control, and transportation. References [14, 15] proposed Tikhonov regularization method and a general regularization method for solving inverse

variational inequality problems. Luo and Yang [16, 17] further extended the results of [14, 15] to the inverse mixed variational inequality problems. The generalized quasi-variational inequality problem was introduced in [18]. However, to our knowledge, a few works implemented quasi-variational inequality to deal with image restoration problem. Recently, Lenzen et al. [19–21] firstly considered a class of quasi-variational inequalities for studying adaptive image restoration, where adaptivity is solution-driven adaptivity. Moreover, they showed that a lot of experimental results support their theoretical findings.

Inspired and motivated by the works of [19, 20], in this paper, we introduce a general TV regularization which includes TV regularization of the classical ROF model [1] as its special case. For solving the minimization problem with generalized TV regularization, we discuss its dual problem, which is like the following formulation:

$$\min_{p \in \mathcal{D}} \{F(p) + \Phi(p)\}, \quad (1)$$

where \mathcal{D} is a convex constraint set. For generalizing the regularization approach to solution-driven adaptivity, we find a fixed point of the following mapping:

$$p_0 \mapsto \bar{p} := \operatorname{argmin}_{p \in \mathcal{D}(p_0)} \{F(p) + \Phi(p)\}. \quad (2)$$

The above fixed point problem is equivalent to solving a mixed quasi-variational inequality [22]. We provide the existence and uniqueness of a fixed point for the mixed quasi-variational inequality for adaptive image restoration. Thus, our theoretical results generalize the research works of [19]. Meanwhile, we propose a modified projection algorithm for solving mixed quasi-variational inequality and prove its convergence. Finally, we give improved experimental results compared to the experiments presented in [19]. Moreover, our experimental results show that the solution-driven adaptive generalized TV model produces excellent restoration effects for different test images.

The rest of this paper is organized as follows. In Section 2, we recall some notations concerned with generalized Φ -projection operator. In Section 3, we introduce the generalized TV regularization which covers other TV regularizers given in literature. Our model of solution-driven adaptivity described by means of mixed quasi-variational inequalities is shown in Section 4. We consider the theoretical results in Section 5, where we prove the existence and uniqueness of the solution for mixed quasi-variational inequality. In Section 6, we present a modified projection algorithm and its convergence. We give a lot of numerical experiments supporting our theoretical results and showing our better improvement in Section 7. Finally, we conclude this paper in Section 8.

2. Preliminaries

In this section, we recall the concept of the generalized Φ -projection operator, together with its properties.

Let Ω be a nonempty closed convex subset of \mathbb{R}^n . Let $G : \mathbb{R}^n \times \Omega \rightarrow \mathbb{R} \cup \{+\infty\}$ be a function defined as follows:

$$G(x, \xi) = \|x\|^2 - 2\langle x, \xi \rangle + \|\xi\|^2 + 2\rho\Phi(\xi), \quad (3)$$

where $\xi \in \Omega$, $x \in \mathbb{R}^n$, ρ is a positive number, and $\Phi : \Omega \rightarrow \mathbb{R} \cup \{+\infty\}$ is a proper, convex, and lower semicontinuous function.

Definition 1. The generalized Φ -projection operator $\Pi_{\Omega}^{\Phi} : \mathbb{R}^n \rightrightarrows \Omega$ is defined as

$$\Pi_{\Omega}^{\Phi}(x) = \left\{ u \in \Omega : G(x, u) = \inf_{\xi \in \Omega} G(x, \xi) \right\}, \quad (4)$$

$\forall x \in \mathbb{R}^n.$

From Lemmas 3.1 and 3.2 of [25], we know that Π_{Ω}^{Φ} is a single valued and nonexpansive mapping and $x^* = \Pi_{\Omega}^{\Phi}(x)$ if and only if

$$\langle x^* - x, y - x^* \rangle + \rho\Phi(y) - \rho\Phi(x^*) \geq 0, \quad \forall y \in \Omega. \quad (5)$$

3. Generalized TV Regularization

In this section, we introduce a new variational approach for image denoising and nonblind image deblurring that is based on total variation regularization. Our general approach covers various adaptive and anisotropic types of TV regularization approaches.

The image deblurring problem formulation is as follows. Let $f \in \mathbb{R}^n$ be a degraded noisy image, which is obtained from a noise-free image $u \in \mathbb{R}^n$ by convolution with a blurring kernel M , followed by an addition of Gaussian noise; that is,

$$f = Mu + \delta, \quad (6)$$

where M is $n \times n$ invertible matrix and δ is a Gaussian random variable with zero mean. The above problem is a typical inverse problem. In order to recover u from f , assuming that $u \mapsto M * u$ is a mapping from $\mathbb{R}^n \rightarrow \mathbb{R}^n$, we aim at considering the following optimization problem:

$$\operatorname{argmin}_{u \in \mathbb{R}^n} E(u) = \frac{1}{2} \|Mu - f\|_{L^2}^2 + \alpha \operatorname{TV}(u), \quad \alpha > 0. \quad (7)$$

In particular, if $M := \operatorname{Id}$, the minimization problem (7) reduces to image denoising problem, where Id denotes an identity mapping.

Now we denote by $L : \mathbb{R}^{nd} \rightarrow \mathbb{R}^n$ the discretization of the divergence operator div and denote by $\|\cdot\|_{L_2}$ norm. Let us define the following generalized total variation regularizer:

$$\alpha \operatorname{TV}(u) := \sup_{p \in \mathcal{D}} \left\{ \langle Lp, u \rangle_{L_2} - \|M^{-T}Lp\|_2 \right\}, \quad (8)$$

where

$$\mathcal{D} = \{p \in \mathbb{R}^{nd}, p_i \in B_{\alpha}(0), i = 1, \dots, n\} \quad (9)$$

for $p = (p_1, p_2, \dots, p_n)^\top$ with $p_i \in \mathbb{R}^d$ and $B_\alpha(0)$ is d -dimensional closed ball with radius α centered at 0. Then the optimization problem is given as

$$\begin{aligned} \operatorname{argmin}_{u \in \mathbb{R}^n} E(u) &= \frac{1}{2} \|Mu - f\|_2^2 \\ &+ \sup_{p \in \mathcal{D}} \left\{ \langle Lp, u \rangle_{L_2} - \|M^{-\top} Lp\|_2 \right\}. \end{aligned} \quad (10)$$

Problem (10) includes a large variety of problems as its special cases:

- (i) If $\|M^{-\top} Lp\|_2 = 0$, then (10) reduces to problem (2.12) of [19].
- (ii) If $\|M^{-\top} Lp\|_2 = 0$ and $M := \text{Id}$, then (10) reduces to the classical ROF model of [1].

We derive the corresponding dual problem of (10) as follows. The optimality condition for u reads

$$M^\top (Mu - f) + Lp = 0. \quad (11)$$

It follows from (11) that

$$u = M^{-1} (f - M^{-\top} Lp), \quad (12)$$

$$Mu = f - M^{-\top} Lp, \quad (13)$$

where $M^{-\top} := (M^\top)^{-1}$. Using the abbreviation $A := M^{-\top} L$, from (11) and (12) we obtain the dual problem

$$\begin{aligned} E^*(p) &= \frac{1}{2} \|Mu - f\|_2^2 + \langle Lp, u \rangle_{L_2} - \|M^{-\top} Lp\|_2 \\ &= \frac{1}{2} \|M^{-\top} Lp\|_2^2 + f^\top M^{-\top} Lp \\ &\quad - (M^{-1} M^{-\top} Lp)^\top Lp - \|M^{-\top} Lp\|_2 \\ &= \frac{1}{2} \|Ap\|_2^2 + fAp - (Ap)^\top Ap - \|Ap\|_2 \\ &= -\frac{1}{2} (\|Ap\|_2^2 - 2fAp) - \|Ap\|_2 \\ &= -\frac{1}{2} (\langle Ap - f, Ap - f \rangle_{L_2} - \|f\|_2^2) - \|Ap\|_2 \\ &= -\frac{1}{2} \|Ap - f\|_2^2 + \frac{1}{2} \|f\|_2^2 - \|Ap\|_2. \end{aligned} \quad (14)$$

When maximizing $E^*(p)$ over \mathcal{D} , the constant term $(1/2)\|f\|_2^2$ can be omitted without changing the optimum. Moreover, the maximization of E^* equals the minimization of $G(p) := -E^*(p)$, and we can formulate the dual problem of (10) as

$$\begin{aligned} \operatorname{argmin}_{p \in \mathcal{D}} G(p) &= F(p) + \Phi(p) \\ &= \frac{1}{2} \|Ap - f\|_2^2 + \|Ap\|_2 \end{aligned} \quad (15)$$

with $\mathcal{D} := \mathcal{D}_{\text{loc}}^1 \times \mathcal{D}_{\text{loc}}^2 \times \dots \times \mathcal{D}_{\text{loc}}^n$, where each local constraint set $\mathcal{D}_{\text{loc}}^i$ is a d -dimensional closed ball.

From a solution \bar{p} of the dual problem, we can retrieve the solution \bar{u} of the primal problem by $\bar{u} = M^{-1}(f - A\bar{p})$. Therefore, the key issue in our discussion of generalized TV image restoration problem is to solve the above minimization problem (15).

4. Solution-Driven Adaptivity

In [20], Lenzen et al. proposed a kind of adaptivity, where the constraint set \mathcal{D} depends on the unknown solution p of the problem. Naturally, the adaptivity is determined by the noise-free image u , which can be obtained by $u = M^{-1}(f - Ap)$. Moreover, the experimental results of [19] showed that the adaptivity was improved by solution-driven model. In the following, in order to study generalized TV image restoration with solution-driven adaptivity, we generalize problem (15) by introducing a dependency of \mathcal{D} on the dual variable: find a fixed point p^* of the mapping

$$p_0 \mapsto \bar{p} := \operatorname{argmin}_{p \in \mathcal{D}(p_0)} G(p) = F(p) + \Phi(p). \quad (16)$$

Having found a fixed point p^* , the corresponding constraint set is $\mathcal{D}(p^*)$; that is, the adaptivity is solution-driven. Solving problem (16) is equivalent to considering the following mixed quasi-variational inequality (MQVI):

$$\begin{aligned} \text{find } p^* \in \mathcal{D}(p^*) \text{ such that } &\langle \nabla F(p^*), p - p^* \rangle \\ &+ \Phi(p) - \Phi(p^*) \geq 0, \quad \forall p \in \mathcal{D}(p^*), \end{aligned} \quad (17)$$

where $F(p) = (1/2)\|Ap - f\|_2^2$ and $\Phi(p) = \|Ap\|_2$.

From (5), it is easy to see that the MQVI (17) is equivalent to the following projection equation:

$$p^* = \Pi_{\mathcal{D}(p^*)}^\Phi(p^* - \rho \nabla F(p^*)), \quad \forall \rho > 0. \quad (18)$$

5. Theory for MQVI

In this section, we provide the existence and uniqueness results for the MQVI (17).

5.1. Existence of Solutions

Theorem 2. Let $F(p) := (1/2)\|Ap - f\|_2^2$ and $\Phi(p) := \|Ap\|_2$, where $A : \mathbb{R}^{mn} \rightarrow \mathbb{R}^n$ is a linear operator. Let $\mathcal{D}(p)$ be defined as follows:

$$\mathcal{D} : \bar{p} \rightrightarrows \mathcal{D}(\bar{p}) := \{p \in \mathbb{R}^{mn} : p_i \in \mathcal{D}_{\text{loc}}^i(\bar{p}) \subset \mathbb{R}^{mn}, i = 1, \dots, n\}, \quad (19)$$

where each $\mathcal{D}_{\text{loc}}^i : \mathbb{R}^{mn} \rightrightarrows \mathbb{R}^{mn}$, $i = 1, \dots, n$, has the following properties:

- (i) For fixed p the set $\mathcal{D}_{\text{loc}}^i(p)$ is a closed convex subset of \mathbb{R}^{mn} .
- (ii) There exists $C > 0$ such that, for all i, p , one has $\mathcal{D}_{\text{loc}}^i(p) \subset B_C(0)$.

- (iii) There exists $c > 0$ such that, for every p and every i , one has $B_c(0) \subset \mathcal{D}_{loc}^i(p)$. In particular, $\mathcal{D}_{loc}^i(p)$ is nonempty.
- (iv) The generalized Φ -projection operator $\Pi_{\mathcal{D}_{loc}^i(p)}^\Phi(q)$ of q onto $\mathcal{D}_{loc}^i(p)$ for a fixed q is continuous with respect to p .

Then mixed quasi-variational inequality (17) has a solution.

Proof. Firstly, we know from the definition of $\mathcal{D}(p)$ that

$$\mathcal{D}(p) := \mathcal{D}_{loc}^1(p) \times \mathcal{D}_{loc}^2(p) \times \cdots \times \mathcal{D}_{loc}^n(p). \quad (20)$$

Thus, from assumption (ii), we immediately derive that

$$\mathcal{D}(p) \subset (B_C(0))^n \subset B_{\sqrt{n}C}(0), \quad p \in \mathbb{R}^{mn}. \quad (21)$$

Since $B_{\sqrt{n}C}(0)$ is a bounded closed convex ball, $B_{\sqrt{n}C}(0)$ is compact convex.

Assumptions (i) and (iii) imply that \mathcal{D} is a nonempty closed convex valued mapping on $B_{\sqrt{n}C}(0)$. Moreover, $\Phi(p) = \|Ap\|_2$ is a proper, convex, and lower semicontinuous function. Hence the generalized Φ -projection operator $\Pi_{\mathcal{D}(p)}^\Phi$ is well defined. By $\Pi_{\mathcal{D}(p)}^\Phi(q) = (\Pi_{\mathcal{D}_{loc}^1(p)}^\Phi(q_1), \dots, \Pi_{\mathcal{D}_{loc}^n(p)}^\Phi(q_n))^\top$ and (iv), we obtain that $\Pi_{\mathcal{D}(p)}^\Phi$ is continuous with respect to p .

Now we define the mapping $S : B_{\sqrt{n}C}(0) \rightarrow B_{\sqrt{n}C}(0)$ by

$$S(p) := \Pi_{\mathcal{D}(p)}^\Phi(p - \rho \nabla F(p)), \quad \forall \rho > 0. \quad (22)$$

It follows from the continuity of $\nabla F(p) = A^\top(Ap - f)$ and $\Pi_{\mathcal{D}(p)}^\Phi$ that S is continuous. Hence by (18) and the Brouwer fixed point theorem, we have that problem (17) has a solution. \square

In the case that $\Phi \equiv 0$, a similar result of Theorem 2 was obtained (see Proposition 4.2 of [19]). Therefore, Theorem 2 can also be considered as a generalization of Proposition 4.2 of [19].

5.2. Uniqueness Result of the Proposed Approach. In this subsection, let us consider the uniqueness results for MQVI (17). In [19], Lenzen et al. discussed the uniqueness of solution of quasi-variational inequality on only a subspace of \mathbb{R}^{mn} , because ∇F is not strongly monotone on the null space $\mathcal{N}(A)$ of A ($\mathcal{N}(A) := \{x \in \mathbb{R}^n : Ax = 0\}$). On the other hand, our main aim is to find $u = M^{-1}(f - Ap)$, which does not depend on the component of p in $\mathcal{N}(A)$. In view of these reasons mentioned, [19] restricted the problem of quasi-variational inequality to the complement $\mathcal{N}^\perp(A)$ of $\mathcal{N}(A)$. Now we utilize similar method to study problem (17). Firstly, we give the following search model that is restricted to $\mathcal{N}^\perp(A)$:

$$\text{Find } p^* \in \Pi_{\mathcal{N}^\perp(A)} \mathcal{D}(p^*) \quad (23)$$

$$\text{such that } \langle \nabla F(p^*), p - p^* \rangle + \Phi(p) - \Phi(p^*) \geq 0, \quad (24)$$

$$\forall p \in \Pi_{\mathcal{N}^\perp(A)} \mathcal{D}(p^*).$$

Theorem 3. Assume that the set $\mathcal{D}(p)$ depends only on $p_{res} := \Pi_{\mathcal{N}^\perp(A)}(p)$.

- (i) Let p_{res}^* be a solution to the restricted problem (23), and then any $p^* \in \Pi_{\mathcal{N}^\perp(A)}^{-1}(p_{res}^*) \cap \mathcal{D}(p_{res}^*)$ is a solution to the original problem (17).
- (ii) Let p^* be a solution to the unrestricted problem (17), and then any $p_{res}^* \in \Pi_{\mathcal{N}^\perp(A)}^{-1}(p^*)$ is a solution to the restricted problem (23).

Proof. Let $p_{res}^* \in \Pi_{\mathcal{N}^\perp(A)}(\mathcal{D}(p_{res}^*))$ be a solution to the restricted problem (23); that is,

$$\langle \nabla F(p_{res}^*), p - p_{res}^* \rangle + \Phi(p) - \Phi(p_{res}^*) \geq 0, \quad (25)$$

$$\forall p \in \Pi_{\mathcal{N}^\perp(A)}(\mathcal{D}(p_{res}^*)).$$

For any $p^* \in \mathcal{D}(p_{res}^*)$ such that $p_{res}^* = \Pi_{\mathcal{N}^\perp(A)} p^* = p^* - \Pi_{\mathcal{N}(A)} p^*$, it holds that $p^* \in \mathcal{D}(p^*) = \mathcal{D}(p_{res}^*)$. Now let $p \in \mathcal{D}(p_{res}^*) = \mathcal{D}(p^*)$ be arbitrary. We decompose p into $p = p_{res} + p_{\mathcal{N}}$, where $p_{res} := \Pi_{\mathcal{N}^\perp(A)} p$ and $p_{\mathcal{N}} := \Pi_{\mathcal{N}(A)} p$. Then it follows from (25) and $Ap = Ap_{res}$ and $Ap^* = Ap_{res}^*$ that

$$\begin{aligned} & \langle \nabla F(p^*), p - p^* \rangle + \Phi(p) - \Phi(p^*) \\ &= \langle A^\top(Ap^* - f), p - p^* \rangle + \|Ap\|_2 - \|Ap^*\|_2 \\ &= \langle Ap^* - f, A(p - p^*) \rangle + \|Ap\|_2 - \|Ap^*\|_2 \\ &= \langle Ap_{res}^* - f, A(p_{res} - p_{res}^*) \rangle + \|Ap_{res}\|_2 \\ & \quad - \|Ap_{res}^*\|_2 \\ &= \langle A^\top(Ap_{res}^* - f), p_{res} - p_{res}^* \rangle + \|Ap_{res}\|_2 \\ & \quad - \|Ap_{res}^*\|_2 \geq 0. \end{aligned} \quad (26)$$

Thus, p^* is a solution of (17).

Let p^* be a solution of problem (17). In particular, $p^* \in \mathcal{D}(p^*)$. We consider the decomposition $p^* = p_{res}^* + p_{\mathcal{N}}^*$, where $p_{res}^* := \Pi_{\mathcal{N}^\perp(A)} p^*$ and $p_{\mathcal{N}}^* := \Pi_{\mathcal{N}(A)} p^*$. Then

$$p_{res}^* \in \Pi_{\mathcal{N}^\perp(A)}(\mathcal{D}(p^*)) = \Pi_{\mathcal{N}^\perp(A)}(\mathcal{D}(p_{res}^*)). \quad (27)$$

Let $p_{res} \in \Pi_{\mathcal{N}^\perp(A)}(\mathcal{D}(p_{res}^*))$ be arbitrary. There exists $p \in \mathcal{D}(p_{res}^*)$ such that

$$p_{res} = \Pi_{\mathcal{N}^\perp(A)} p = p - \Pi_{\mathcal{N}(A)} p. \quad (28)$$

It follows from (28) that

$$\begin{aligned} & \langle \nabla F(p_{res}^*), p_{res} - p_{res}^* \rangle + \Phi(p_{res}) - \Phi(p_{res}^*) \\ &= \langle Ap_{res}^* - f, A(p_{res} - p_{res}^*) \rangle + \|Ap_{res}\|_2 \\ & \quad - \|Ap_{res}^*\|_2 \\ &= \langle Ap^* - f, A(p - p^*) \rangle + \|Ap\|_2 - \|Ap^*\|_2 \\ &= \langle A^\top(Ap^* - f), p - p^* \rangle + \|Ap\|_2 - \|Ap^*\|_2 \\ & \geq 0, \end{aligned} \quad (29)$$

where the last inequality holds since $p \in \mathcal{D}(p^*)$ due to $\mathcal{D}(p^*) = \mathcal{D}(p_{res}^*)$ and p^* solves (17). Thus p_{res}^* is a solution of (23). \square

From Section 3, we can see that the final purpose of finding the existence and uniqueness of p^* is to solve the optimal problem (10), because $u^* = M^{-1}(f - Ap^*)$ which does not depend on p^* . We therefore focus on $v^* = Ap^*$, which depends only on the component p_{res}^* of p^* in $\mathcal{N}^\perp(A)$; that is, we only need to consider the restricted problem (23). Based on Theorem 3, the restricted problem has a solution if and only if the original problem has a solution. Here we specify $\|v^*\|_2 = \|p^*\|_B := \sqrt{(p^*)^\top A^\top Ap^*}$. In the following discussion, $\widetilde{\mathcal{D}}(v)$ denotes a nonempty, closed, convex set such that $\widetilde{\mathcal{D}}(v) = \mathcal{D}(p) = \widetilde{\mathcal{D}}(Ap)$.

Before showing the uniqueness, let us define $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ such that $\bar{p} \in T(v)$ if and only if $\bar{p} \in \widetilde{\mathcal{D}}(v)$ and \bar{p} is a solution to the following MQVI:

$$\langle \nabla F(\bar{p}), p - \bar{p} \rangle + \Phi(p) - \Phi(\bar{p}) \geq 0, \quad \forall p \in \widetilde{\mathcal{D}}(v). \quad (30)$$

Theorem 4. *Under all the assumption conditions of Theorem 2 and the assumptions*

(i) ∇F is Lipschitz continuous with Lipschitz constant $\mu_2 := \|A^\top A\|_2 > 0$, that is,

$$\|\nabla F(x) - \nabla F(y)\|_2 \leq \mu_2 \|x - y\|_2, \quad \forall x, y \in \mathbb{R}^m, \quad (31)$$

(ii) the generalized Φ -projection operator $\Pi_{\widetilde{\mathcal{D}}(v)}^\Phi q$ is Lipschitz continuous with respect to v with the variation rate $\tilde{\eta} > 0$, that is,

$$\|\Pi_{\widetilde{\mathcal{D}}(v)}^\Phi q - \Pi_{\widetilde{\mathcal{D}}(v')}^\Phi q\|_2 \leq \tilde{\eta} \|v - v'\|_2, \quad \forall q \in \mathbb{R}^m, \quad (32)$$

(iii) $\tilde{\eta} < 1/\sqrt{\mu_2}$,

if p^* is a solution of mixed quasi-variational inequality (17), then $v^* = Ap^*$ is unique.

Proof. Fix $v_1, v_2 \in \text{Im}(A)$. Let $\mathcal{D}_i := \widetilde{\mathcal{D}}(v_i)$, $p_i \in T(v_i)$:

(i) If $A(p_1 - p_2) = 0$, we obtain

$$\begin{aligned} \|AT(v_1) - AT(v_2)\|_2 &= \|Ap_1 - Ap_2\|_2 = 0 \\ &< \|v_1 - v_2\|_2. \end{aligned} \quad (33)$$

(ii) If $A(p_1 - p_2) \neq 0$, since p_i , $i = 1, 2$, solve $\arg \min_{p \in \widetilde{\mathcal{D}}(v_i)} (1/2)\|Ap - f\|_2^2 + \|Ap\|_2$, the following MQVI

$$\langle \nabla F(p_i), q - p_i \rangle + \Phi(q) - \Phi(p_i) \geq 0, \quad \forall q \in \mathcal{D}_i, \quad (34)$$

holds; that is, for any $\rho \geq 0$,

$$p_i = \Pi_{\mathcal{D}_i}^\Phi(p_i - \rho \nabla F(p_i)). \quad (35)$$

In particular, $q := \Pi_{\mathcal{D}_2}^\Phi(p_1 - \rho \nabla F(p_1))$, and it follows from (35) and (ii) that

$$\begin{aligned} \|p_1 - q\|_2 &= \|\Pi_{\mathcal{D}_1}^\Phi(p_1 - \rho \nabla F(p_1)) - \Pi_{\mathcal{D}_2}^\Phi(p_1 - \rho \nabla F(p_1))\|_2 \\ &\leq \tilde{\eta} \|v_1 - v_2\|_2. \end{aligned} \quad (36)$$

On the other hand, $p_2 \in \mathcal{D}_2$ implies that

$$\langle q - (p_1 - \rho \nabla F(p_1)), p_2 - q \rangle + \Phi(p_2) - \Phi(q) \geq 0. \quad (37)$$

Therefore, from (34) and (37), we have

$$\begin{aligned} \langle q - p_1, p_2 - q \rangle &\geq \rho \langle \nabla F(p_1), q - p_2 \rangle - \Phi(p_2) \\ &\quad + \Phi(q) \\ &= \rho \langle \nabla F(p_1), q - p_1 \rangle \\ &\quad + \rho \langle \nabla F(p_2), p_1 - q \rangle \\ &\quad + \rho \langle \nabla F(p_2), q - p_2 \rangle + \Phi(q) \\ &\quad - \Phi(p_2) \\ &\quad + \rho \langle \nabla F(p_1) - \nabla F(p_2), p_1 - p_2 \rangle \\ &\geq \rho \langle \nabla F(p_1) - \nabla F(p_2), q - p_1 \rangle \\ &\quad + \rho \langle \nabla F(p_1) - \nabla F(p_2), p_1 - p_2 \rangle \\ &= \rho \langle A^\top A(p_1 - p_2), q - p_1 \rangle \\ &\quad + \rho \|Ap_1 - Ap_2\|_2^2; \end{aligned} \quad (38)$$

that is,

$$\begin{aligned} \rho \|Ap_1 - Ap_2\|_2^2 &\leq \langle q - p_1, p_2 - q \rangle \\ &\quad + \rho \langle A^\top A(p_1 - p_2), p_1 - q \rangle \\ &\leq \|q - p_1\|_2 \cdot \|q - p_2\|_2 \\ &\quad + \rho \|Ap_1 - Ap_2\|_2 \cdot \|Ap_1 - Aq\|_2 \end{aligned} \quad (39)$$

and, by dividing by $\rho \|Ap_1 - Ap_2\|_2 > 0$, because $Ap_1 - Ap_2 \neq 0$,

$$\begin{aligned} \|Ap_1 - Ap_2\|_2 &\leq \frac{\|q - p_1\|_2 \cdot \|q - p_2\|_2}{\rho \|Ap_1 - Ap_2\|_2} \\ &\quad + \|Ap_1 - Aq\|_2. \end{aligned} \quad (40)$$

Since ρ is arbitrarily large, we find

$$\|Ap_1 - Ap_2\|_2 \leq \|Ap_1 - Aq\|_2. \quad (41)$$

By (i), we have $\|A\|_2 = \sqrt{\mu_2}$. Thus, it follows from (36) that

$$\|Ap_1 - Aq\|_2 \leq \|A\|_2 \|p_1 - q\|_2 \leq \sqrt{\mu_2} \tilde{\eta} \|v_1 - v_2\|_2; \quad (42)$$

that is,

$$\|AT(v_1) - AT(v_2)\|_2 \leq \sqrt{\mu_2} \tilde{\eta} \|v_1 - v_2\|_2. \quad (43)$$

Therefore, we can see from (iii) that $A \circ T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a contractive mapping. Moreover, the Banach fixed point theorem implies that there exists a unique fixed point $v^* = Ap^*$ of $A \circ T$ in \mathbb{R}^n . \square

We already mentioned that, in the considered applications for image restoration, we are actually interested in the variable $u := M^{-1}(f - Ap)$. Obviously, it follows from Theorem 4 that u is unique.

```

Input: The maximal number of iterations  $N$ ; the constraint set  $\mathcal{D}(p^0)$ ; the starting point
 $p^0 \in \mathcal{D}(p^0) \subset B_{\sqrt{\mu}C}(0)$ ; the parameter number  $\rho$  and such that  $0 < \rho < 1/\mu_2$ .
Output:  $p^N$ .
begin
  for  $k = 0$  to  $N - 1$  do
    Step 1.  $\bar{p}^{k-1} := \Pi_{\mathcal{D}(p^{k-1})}^\Phi(p^{k-1} - \rho \nabla F(p^{k-1}))$ ;
    Step 2.  $p^k := \Pi_{\mathcal{D}(\bar{p}^{k-1})}^\Phi(p^{k-1} - \rho \nabla F(\bar{p}^{k-1}))$ ;
    Step 3. If  $|p^k - p^{k-1}| \leq \varepsilon$ , for all  $\varepsilon > 0$ , then, stop; otherwise,  $k = k + 1$  and go the Step 1.
  end
end

```

ALGORITHM 1: Modified projection method for solving the MQVI (17).



FIGURE 1: (a) Noisy input data, MSSIM = 0.327. (b) Solution-driven adaptive TV, MSSIM = 0.812. (c) Solution-driven adaptive generalized TV, MSSIM = 0.822.

6. Numerics

Throughout this section, assume that all the assumption conditions of Theorems 2 and 4 are satisfied. Next, we propose an iterative algorithm to solve the MQVI (17) and show convergence of the proposed algorithm.

6.1. *Proposed Iterative Algorithm.* See Algorithm 1.

6.2. *Convergence of Algorithm 1*

Theorem 5. *Let all the assumptions of Theorems 2 and 4 be satisfied. Moreover, assume that ∇F is monotone; that is,*

$$\langle \nabla F(x) - \nabla F(y), x - y \rangle \geq 0, \quad \forall x, y \in \mathbb{R}^{mn}. \quad (44)$$

Then the sequence $\{p^k\}$ generated by Algorithm 1 converges to the unique solution p^ of MQVI (17).*

Proof. For the proof, see Theorem 2.5 in [26]. \square

7. Experiment Results

7.1. *Improvement of Solution-Driven Adaptive TV Regularization.* In this section, we show that the adaptivity is improved by switching from a solution-driven adaptive TV regularization to a solution-driven adaptive generalized TV model. To

this end, we consider the image Cameraman. We generate test data for the denoising problem by adding Gaussian noise with zero mean and standard deviation 0.1 and for the deblurring problem by applying a blurring operator and adding Gaussian noise with zero mean and standard deviation 0.01. For comparison, we make use of a mean SSIM (MSSIM) index [27] to evaluate the restored image quality in Figures 1 and 2. The definitions of the similarity measures SSIM and MSSIM are given as follows:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_x\sigma_y + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (45)$$

where μ_x, μ_y are the average values of two signals x and y , σ_x^2 and σ_y^2 are the variances of x and y , and c_1, c_2 are variables to stabilize the division with weak denominator

$$\text{MSSIM}(X, Y) = \frac{1}{M} \sum_{j=1}^M \text{SSIM}(x_j, y_j), \quad (46)$$

where X and Y are the reference and the distorted images, respectively; x_j and y_j are the image contents at the j th local window; and M is the number of local windows of the image.

In Figure 1, we give the close-up of the results of denoising the image Cameraman with TV regularization and generalized TV regularization approaches. The values



FIGURE 2: (a) Blurred input data, $MSSIM = 0.032$. (b) Solution-driven adaptive TV, $MSSIM = 0.866$. (c) Solution-driven adaptive generalized TV, $MSSIM = 0.868$.



FIGURE 3: Test images. (a) Cameraman. (b) Lena. (c) Pepper. (d) Building.

for similarity given show that generalized TV regularization approach enhances the reconstruction compared to the TV regularization.

In Figure 2, close-up of the results of deblurring the image Cameraman with TV regularization and generalized TV regularization approaches is shown. In terms of similarity to the original data, we can see that generalized TV regularization

approach improves the reconstruction compared to the TV regularization.

7.2. Comparison of Image Restoration for Different Test Images. In this subsection, we report that our method can remove Gaussian noise efficiently while preserving details very well. For noise removal, we compare our method with some



FIGURE 4: (a) Noisy image (PSNR = 8.748 dB). (b) Local TV, Split Bregman (PSNR = 28.18 dB). (c) Nonlocal TV, Split Bregman (PSNR = 30.35 dB). (d) Anisotropic TV, Split Bregman (PSNR = 27.687 dB). (e) Isotropic TV, Split Bregman (PSNR = 28.594 dB). (f) Solution-driven adaptive generalized TV, modified projection (PSNR = 30.56 dB).

recently developed models, namely, the local TV [28], the NLTV [29], the anisotropic TV [30], and the isotropic TV [30], when the Split Bregman method is considered. The quality of our restoration is measured by the peak signal-to-noise ratio (PSNR) in decibels (dB):

$$\text{PSNR} = 10 \log_{10} \left(\frac{M \times N}{\sum_{i=1}^M \sum_{j=1}^N (u_{i,j} - \tilde{u}_{i,j})^2} \right), \quad (47)$$

where $M \times N$ indicates the image size and u and \tilde{u} denote the original image and the restored one, respectively. Generally, the higher PSNR values indicate better quality of the restored images.

The test images are shown in Figure 3.

In Figure 4, we make use of the local TV model, the NLTV model, the anisotropic TV model, the isotropic TV model, and the solution-driven adaptive generalized TV model to display the restoration results for the test image (a) corrupted by 8% Gaussian noise. Here, our proposed model performs significantly better and can suppress the Gaussian noise successfully.

Table 1 lists the restoration results in the PSNR of different methods for test images Cameraman, Lena, Pepper, and Building corrupted by Gaussian noise.

TABLE 1: Comparison of restoration results in PSNR (dB) for images corrupted by Gaussian noise.

| Methods | Cameraman | Lena | Pepper | Building |
|----------------|--------------|--------------|--------------|-------------|
| Local TV | 28.18 | 27.68 | 26.68 | 27.55 |
| NLTV | 30.35 | 29.87 | 30.86 | 29.23 |
| Anisotropic TV | 27.687 | 26.1 | 26.591 | 29.3 |
| Isotropic TV | 28.594 | 27.902 | 27.661 | 26.603 |
| Proposed | 30.56 | 30.21 | 30.98 | 29.5 |

From Table 1, it is apparent that our proposed method generates the best restoration results for all the test images. Actually, in Table 1, the proposed method obtains the highest PSNR values for all the test images. This demonstrated that our method is more robust for images corrupted by Gaussian noise.

7.3. Expanded Experiments. Remote sensing very often deals with inverting a forward model. To this aim, one has to produce an accurate and robust model able to predict physical, chemical, geological, or atmospheric parameters from spectra, such as surface temperature, water vapour, and ozone; see, for example, [31]. Denoising images could be part

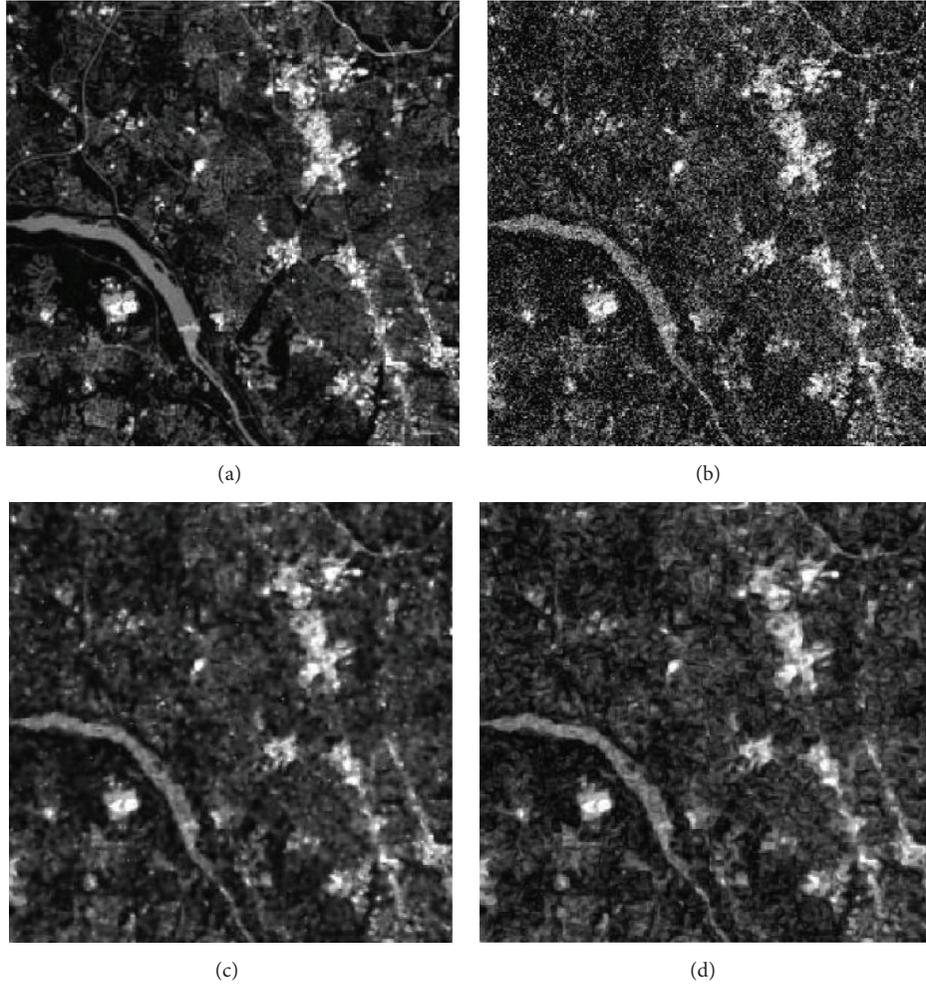


FIGURE 5: (a) Original satellite image. (b) Gaussian noise ($\sigma = 0.02$) image. (c) Fourth-order PDE model. (d) Proposed method.

of the accurate and robust model. In particular, our image restoration method could be used to remove noises in the class of satellite images.

In Figure 5, we give the close-up of the results of denoising the satellite image adding Gaussian noise with the standard deviation $\sigma = 0.02$. Though the fourth-order PDE model removes noise more thoroughly, texture detail also had been changed. Comparing rivers and streets with those in the original image, some information was lost. Our method not only suppresses the noise very well but also keeps a lot of detail and texture information of the original image.

From Table 2, we can see that our method has better denoising effects than pure anisotropic diffusion and fourth-order PDE.

8. Conclusion

In this paper, our main contribution is to introduce the generalized TV regularization which includes various types of TV regularization as special cases. We studied the existence and uniqueness of the solution for mixed quasi-variational

TABLE 2: Comparison of restoration results in PSNR (dB) for satellite images.

| Methods | $\sigma = 0.01$ | $\sigma = 0.02$ | $\sigma = 0.05$ | $\sigma = 1$ |
|---------------------------------|-----------------|-----------------|-----------------|--------------|
| Pure anisotropic diffusion [23] | 30.16 | 23.68 | 15.36 | 6.10 |
| Fourth-order PDE [24] | 31.64 | 25.40 | 17.40 | 11.36 |
| Proposed | 34.32 | 27.68 | 19.54 | 13.28 |

inequality for solution-driven adaptive image denoising and image deblurring with generalized TV regularization. The convergence of proposed algorithm for MQVI was proposed. Moreover, our experimental results showed that we improve the image restoration quality and apply the proposed model to deal with the images in many different fields. Our further work will consider utilizing the theory of other variational inequalities for solving optimization problems with other kinds of generalized TV regularizations.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grants 61379019 and 11401493), Innovation Team Funds of Southwest University for Nationalities (Grant 14CXTD03), Innovative Research Team of the Education Department of Sichuan Province (Grant 15TD0050), The Science and Technology Department of Sichuan Province (Grant 2015JY0027), and Fundamental Research Funds for the Central Universities, Southwest University for Nationalities (Grant 2014NZYQN29).

References

- [1] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1–4, pp. 259–268, 1992.
- [2] J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher, "Structure-texture image decomposition—modeling, algorithms, and parameter selection," *International Journal of Computer Vision*, vol. 67, no. 1, pp. 111–136, 2006.
- [3] I. Bayram and M. E. Kamasak, "A directional total variation," in *Proceedings of the 20th European Signal Processing Conference (EUSIPCO '12)*, pp. 265–269, IEEE, Piscataway, NJ, USA, August 2012.
- [4] B. Berkels, M. Burger, M. Droske, O. Nemitz, and M. Rumpf, "Cartoon extraction based on anisotropic image classification," in *Vision, Modeling, and Visualization Proceedings*, pp. 293–300, IOS, Amsterdam, The Netherlands, 2006.
- [5] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation. A SIAM Interdisciplinary Journal*, vol. 4, no. 2, pp. 490–530, 2005.
- [6] G. Gilboa and S. Osher, "Nonlocal operators with applications to image processing," *Multiscale Modeling & Simulation*, vol. 7, no. 3, pp. 1005–1028, 2008.
- [7] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, 2010.
- [8] S. Setzer, G. Steidl, and T. Teuber, "Infimal convolution regularizations with discrete l1-type functionals," *Communications in Mathematical Sciences*, vol. 9, no. 3, pp. 797–827, 2011.
- [9] F. Lenzen, F. Becker, and J. Lellmann, "Adaptive second-order total variation: an approach aware of slope discontinuities," in *Scale Space and Variational Methods in Computer Vision: 4th International Conference, SSVM 2013, Schloss Seggau, Leibnitz, Austria, June 2–6, 2013. Proceedings*, vol. 7893 of *Lecture Notes in Computer Science*, pp. 61–73, Springer, New York, NY, USA, 2013.
- [10] 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.
- [11] T. Chan, S. Esedoglu, F. Park, and A. Yip, "Total variation image restoration: overview and recent developments," in *Handbook of Mathematical Models in Computer Vision*, N. Paragios, Y. Chen, and O. D. Faugeras, Eds., vol. 17, chapter 2, pp. 17–31, Springer, New York, NY, USA, 2006.
- [12] J. P. Oliveira, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Adaptive total variation image deblurring: a majorization-minimization approach," *Signal Processing*, vol. 89, no. 9, pp. 1683–1693, 2009.
- [13] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Springer, New York, NY, USA, 2003.
- [14] X.-P. Luo, "Tikhonov regularization methods for inverse variational inequalities," *Optimization Letters*, vol. 8, no. 3, pp. 877–887, 2014.
- [15] X.-P. Luo and J. Yang, "Regularization and iterative methods for monotone inverse variational inequalities," *Optimization Letters*, vol. 8, no. 4, pp. 1261–1272, 2014.
- [16] X.-p. Luo and H. Yang, "Regularization and iterative methods for inverse mixed variational inequalities," *Nonlinear Analysis Forum*, vol. 19, pp. 53–63, 2014.
- [17] X.-P. Luo and H. Yang, "Tikhonov regularization methods for inverse mixed variational inequalities," *Advances in Nonlinear Variational Inequalities*, vol. 17, no. 2, pp. 13–25, 2014.
- [18] D. Chan and J.-S. Pang, "The generalized quasi-variational inequality problem," *Mathematics of Operations Research*, vol. 7, no. 2, pp. 211–222, 1982.
- [19] F. Lenzen, J. Lellmann, F. Becker, and C. Schnörr, "Solving quasi-variational inequalities for image restoration with adaptive constraint sets," *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, pp. 2139–2174, 2014.
- [20] F. Lenzen, F. Becker, J. Lellmann, S. Petra, and C. Schnörr, "A class of quasi-variational inequalities for adaptive image denoising and decomposition," *Computational Optimization and Applications*, vol. 54, no. 2, pp. 371–398, 2013.
- [21] F. Lenzen, F. Becker, J. Lellmann, S. Petra, and C. Schnörr, "Variational image denoising with adaptive constraint sets," in *Scale Space and Variational Methods in Computer Vision: Third International Conference, SSVM 2011, Ein-Gedi, Israel, May 29–June 2, 2011, Revised Selected Papers*, vol. 6667 of *Lecture Notes in Computer Science*, pp. 206–217, Springer, New York, NY, USA, 2012.
- [22] M. A. Noor and K. I. Noor, "Some new classes of extended general mixed quasi-variational inequalities," *Abstract and Applied Analysis*, vol. 2012, Article ID 962978, 9 pages, 2012.
- [23] L. Alvarez and J. M. Morel, "Formalization and computational aspects of image analysis," *Acta Numerica*, vol. 3, pp. 1–59, 1994.
- [24] 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.
- [25] X. Li, X. S. Li, and N. J. Huang, "A generalized f-projection algorithm for inverse mixed variational inequalities," *Optimization Letters*, vol. 8, no. 3, pp. 1063–1076, 2014.
- [26] A. Nagurney, *Network Economics. A Variational Inequality Approach*, Kluwer Academic Publishers, Boston, Mass, USA, 1999.
- [27] 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.
- [28] T. Goldstein and S. Osher, "The split Bregman algorithm for L1 regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.

- [29] X. W. Liu and L. H. Huang, "A new nonlocal total variation regularization algorithm for image denoising," *Mathematics and Computers in Simulation*, vol. 97, pp. 224–233, 2014.
- [30] Z. W. Qin, D. Goldfarb, and S. Q. Ma, "An alternating direction method for total variation denoising," *Optimization Methods & Software*, vol. 30, no. 3, pp. 594–615, 2015.
- [31] L. Gómez-Chova, J. Muñoz-Marí, V. Laparra, J. Malo-López, and G. Camps-Valls, "A review of kernel methods in remote sensing data analysis," in *Optical Remote Sensing*, vol. 3, pp. 171–206, Springer, Berlin, Germany, 2011.

Research Article

Hybrid Predictor and Field-Biased Context Pixel Selection Based on PPVO

Hongyin Xiang, Jinsha Yuan, and Sizu Hou

Institute of Electrical and Electronic Engineering, North China Electric Power University, Baoding 071000, China

Correspondence should be addressed to Hongyin Xiang; leo_xianghy@126.com

Received 26 November 2015; Revised 29 January 2016; Accepted 1 February 2016

Academic Editor: Daniel Zaldivar

Copyright © 2016 Hongyin 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.

Most pixel-value-ordering (PVO) predictors generated prediction-errors including -1 and 1 in a block-by-block manner. Pixel-based PVO (PPVO) method provided a novel pixel scan strategy in a pixel-by-pixel way. Prediction-error bin 0 is expanded for embedding with the help of equalizing context pixels for prediction. In this paper, a PPVO-based hybrid predictor (HPPVO) is proposed as an extension. HPPVO predicts pixel in both positive and negative orientations. Assisted by expansion bins selection technique, this hybrid predictor presents an optimized prediction-error expansion strategy including bin 0 . Furthermore, a novel field-biased context pixel selection is already developed, with which detailed correlations of around pixels are better exploited more than equalizing scheme merely. Experiment results show that the proposed HPPVO improves embedding capacity and enhances marked image fidelity. It also outperforms some other state-of-the-art methods of reversible data hiding, especially for moderate and large payloads.

1. Introduction

Digital watermarking has been used for copyright protection [1, 2], integrity check [3, 4], and self-identity [5]. Important and secret data as watermark was embedded into the host image. However, in some sensitive scenarios, such as medical images, remote sensing images, and military images, even a small amount of deviation is highly detrimental. Reversible data hiding has improved to this demand [6, 7]. Reversibility here means that pixels in marked image can be perfectly recovered to the original ones in the host image.

Three categories of solutions can be summarized to this problem, difference expansion (DE) [8–10], histogram shifting (HS) [11–14], and integer transformation [15–19]. Difference expansion computes and expands the differences between pixels rather than the pixel itself to embed data. This method has considerable capacity and little distortion. However, difference relies on locations on the map occupying spaces large enough for the payload. Histogram shifting uses the peak point bin to carry data and shifts other bins to guarantee reversibility. The peak bin contains the most common pixels. Residual images composed of the difference of adjacent pixels or various neighboring pixels can also be

used to modify the histogram [20]. In this way, gray values are modified at most by 1 . In this way, this method achieves high fidelity. However, it can only be used to hide limited amounts of information. To remedy the above shortages, sorting [21, 22] and prediction-error expansion (PEE) [23] were incorporated into existing methods. Pixel-value-ordering (PVO) in [24] connected these two new stages together. PVO-K in [25] used equivalent multiextreme pixels overlooked by PVO to carry data bits with identical value. Improved PVO (IPVO) in [26] strengthened integrated behavior using a new predictor different from PVO. Notice that all the three aforementioned methods hunt embedding pixels within artificial separated nonoverlapped image blocks. Pixel-based pixel-value-ordering (PPVO) in [27] predictors provided a novel pixel search strategy in a pixel-by-pixel manner. Proximate pixels were selected to form a context pixel vector and predict objective pixel. Experimental results demonstrated markedly higher image quality and larger embedding capacity (EC) than PVO, PVO-K, and IPVO. Nevertheless, only prediction-error 0 was expanded and only some of the context pixel selection strategies have been brought into use. PPVO takes more practical value as an outstanding pixel hunting mode. In this paper, a hybrid predictor is proposed including errors

0 and 1 expanded rather than only error 0 in the previous work of PPVO. Meanwhile, a novel field-biased context pixel selection scheme is here incorporated in for more accurate prediction than equalizing type merely. Furthermore, prediction-error expansion bins selection (EBS) develops an effective mechanism for optimized performance. Experiment results show that the proposed HPPVO may provide higher embedding capacity and better marked image fidelity than conventional PPVO.

The rest of this paper is organized as follows. In Section 2, some related works are briefly described and the primary motivation for improving PPVO is also stated. In Section 3, the concrete scheme of the proposed HPPVO is analyzed in detail. Section 4 lists extensive experiments compared to state-of-the-art studies. Section 5 concludes this paper.

2. Related Work

2.1. Prediction-Error Expansion. Thodi and Rodriguez [23] first proposed PEE method, which predicted pixels and embedded data employing the produced errors instead of pixel differences. This new technique exploited the potential of the correlation inherent in neighborhood pixels. Thus, EC of PEE doubles that of DE-based method. What is more, complex predictors can make much smaller prediction-error for higher fidelity. Afterwards, Li et al. [28] combined adaptive embedding and pixel selection for significant improvement. They tried to adaptively embed data into expanded pixels in the light of different complexity. This technique guaranteed only those small prediction-errors to be applied for expansion and embedding. Experimental results demonstrated that it was an effective way and outperformed some other state-of-the-art methods.

Optimal expansion bins selection was described in some previous works. Wang et al. [29] selected prediction-error expansion in a dynamical way. Embedding distortion was precalculated for all possible embedding pixels and determined by minimal distortion. Hwang et al. [30] upgraded histogram shifting by sorting predicted-errors. This strategy was able to search the optimal threshold values effectively. Wu and Huang [31] modified a pair of histogram bins in the way of selecting histogram bins in sequence. Four prediction modes produced a large amount of prediction-errors from cover image. Ultimately, a number of histogram pairs were enumerated to obtain the best performance after synthesis combination of those four modes. Xuan et al. [32] expanded prediction-errors using histogram pair according to embedding threshold, fluctuation threshold, and left-

and right-histogram shrinking thresholds. The embedding threshold was for selecting prediction-errors not larger than the threshold value. The fluctuation threshold was applied to select prediction-errors, whose associated neighbor fluctuation not exceeding this threshold. The left- and right-histogram shrinking thresholds were used to find the left and right possibly shrink histogram. Just for the condition of satisfied four thresholds, embedding data was carried out.

2.2. PVO-Based Predictors. Recently, based on PEE method, Li et al. [24] presented a high-fidelity data hiding method incorporating a new prediction strategy using pixel-value-ordering. Cover image was partitioned to be equal-sized but nonoverlapped pixel blocks. Pixels in each block were then ordered by gray-scale values. In this pixel sequence, the maximum and minimum pixels were assigned to predict the submaximum and subminimum pixels. Hence, this pixel selection mechanism reduced the number of shifted pixels for deprecating embedding distortion.

Peng et al. [26] pointed that 0 bin was usually the peak in image histogram and took the relative location relationships of pixels in a block. Improved PVO was raised by expanding error 0 for embedding other than positive 1 and negative error -1 in the previous work of PVO. PVO-K came up and Ou et al. [25] took equivalent pixels of all maximum-valued or minimum-valued ones as unit for embedding. Comparing with PVO, in which only two bits can be embedded into each block at most, PVO-K exploited image redundancy better for more allowable bits.

2.3. PPVO Predictor. All aforementioned predictors predicted pixel in a block-by-block manner. To some extent, this mechanism limited the improvement of EC with high PSNR. Qu and Kim [27] raised a novel pixel-based PVO method, with which pixel is predicted in pixel-by-pixel way. This method exploded block constraint and predicted more pixels than the above block-based predictors. Beyond question, PPVO promoted embedding capacity sufficiently. Right here, context pixels, defined to be neighboring pixels inside some scope, assisted to predict the current pixel. Sorting and ordering must be executed first before prediction. Each prediction value was determined by $\max(C)$ or $\min(C)$, the maximum or minimum of context pixel vector C . When $\max(C)$ is different from $\min(C)$, pixel can get prediction if its gray-scale value x is more than $\max(C)$ or smaller than $\min(C)$. When $\max(C)$ is the same as $\min(C)$, equaling a constant number VC , prediction worked only if pixel value is equal to or less than VC . The following formula here is applied to describe this predictor:

$$\hat{x} = \begin{cases} \max(C), & \text{if } \max(C) \neq \min(C), x \geq \max(C), \\ \min(C), & \text{if } \max(C) \neq \min(C), x \leq \min(C), \\ 254, & \text{if } \max(C) = \min(C) = VC, VC = 254, \\ VC, & \text{if } \max(C) = \min(C) = VC, 1 \leq VC \leq 253, x \leq VC. \end{cases} \quad (1)$$

With (1), some conclusions can be drawn as follows:

- (a) Pixels in smooth blocks tend to be predicted. After sorting and ordering, the actual objective becomes to search equivalent or approximate pixels in context pixel vector.
- (b) Different number of context pixels leads to different predicting accuracy, concerned with image distortion. For better embedding performance, as many as possible context pixels should be selected.
- (c) Context pixels locate inside square blocks. Besides the number, context pixel distribution may cause influences on prediction accuracy.

After intensive investigation, we find that when context pixels are the same ones, prediction probability is on the low side for two reasons. Conventional PPVO took 254 as a typical pixel when $VC = 254$, which cut down nearly a half of possible pixel values. Only pixels equal to or smaller than VC can be predicted. The second feature is that only prediction-error 0 can be expanded for carrying data, which wastes some useful errors for expansion and embedding data. Furthermore, there are still some other optimal context pixel selecting strategies which may strengthen data hiding performance in both EC and PSNR. Hence, in next section, a hybrid predictor based on PPVO is proposed.

3. Proposed Method

PPVO generates predictions in a pixel-by-pixel manner. It breaks through block constraints in PVO, PVO-K, and IPVO. Each pixel is predicted by a context vector that contains many sorted neighboring pixels. Huge amounts of pixel redundancy

in smooth image regions that are ignored in other PVO-based methods can be used to embed data. Unfortunately, only prediction-error 0 is involved in prediction. This addresses smooth image region pixels but it does not address common region pixels. Using more prediction values should improve embedding capacity.

3.1. Hybrid Predictor Based on PPVO

3.1.1. Pixel Prediction. For clarity, the same notations used in PPVO were used in this section. In PPVO, pixels other than the current pixel in a dynamic block are considered context pixels for prediction. After sorting in ascending order, a context pixel vector consisting in such neighboring pixels is defined and noted as $C = (c_1, \dots, c_{CN})$. Here, CN is the number of referenced pixels. The current pixel x can be predicted by the maximum $\max(C)$ and minimum $\min(C)$. Pixels are predicted in two different scenarios. One case is that context pixels have distinct numerical values, and if the gray-scale value of the current pixel $x \geq \max(C)$, then the prediction value is $\max(C)$. Similarly, if the current pixel $x \leq \min(C)$, then prediction value is $\min(C)$. If the current pixel value is greater than $\min(C)$ and less than $\max(C)$, no prediction can be made. Another condition is that all the pixels are identical with a constant number VC ; then the number just makes prediction. To prevent overflow and underflow, preprogress must be conducted before prediction. If pixel value is 255 or 0, it must be modified to 254 or 1 accordingly. These pixels should be recorded by a location map preparing for revision at decoder. Hence, this constant number VC is constrained in the range of $[1, 254]$. Obviously, prediction value of the current pixel x is determined by reference pixels in context pixel vector.

The current pixel x acquires prediction value \hat{x} as follows:

$$\hat{x} = \begin{cases} \max(C), & \text{if } \max(C) \neq \min(C), x \geq \max(C), \\ \min(C), & \text{if } \max(C) \neq \min(C), x \leq \min(C), \\ VC + 1, & \text{if } \max(C) = \min(C) = VC, 1 \leq VC \leq 254, x \geq VC + 1, \\ VC, & \text{if } \max(C) = \min(C) = VC, 1 \leq VC \leq 254, x \leq VC. \end{cases} \quad (2)$$

This makes great differences from prediction algorithm in [27] when all the context pixels take the same value. Differences are produced when $\max(C)$ is equal to $\min(C)$, denoted by VC , as illustrated in Figure 1. (a) shows that when $VC = 254$, PPVO predicts the current pixel only if it is 254, and otherwise it is skipped. However, based on that first step of prediction, HPPVO tries further to discover the lower pixel value than 254 and makes prediction. These two steps cover all the possible scenarios. (b) describes that when VC is less than 254 and is more than zero, PPVO employs only one step to estimate the current pixel just using the sole baseline of VC . Meanwhile, HPPVO takes two baselines, $VC + 1$ in the first step and VC in the second step. Obviously, the proposed

hybrid predictor of HPPVO provides more opportunities to get prediction.

Referring to Figure 2, in $(a_1) \sim (a_5)$, PPVO predicts the current pixel only in (a_1) while HPPVO makes predictions in all these five cases. In fact, according to this algorithm, the other 253 cases of the current pixel can all be predicted in HPPVO. In this light, prediction capacity of HPPVO is 254 times that of PPVO. Referring to Figure 2, in $(b_1) \sim (b_5)$, HPPVO still predicts the current pixel in all the cases. Yet, when the current pixel is more than VC , prediction is invalid and has to be skipped in that previous work of PPVO. In this sense, the smaller the VC , the less the prediction cases. So, HPPVO improves obviously prediction

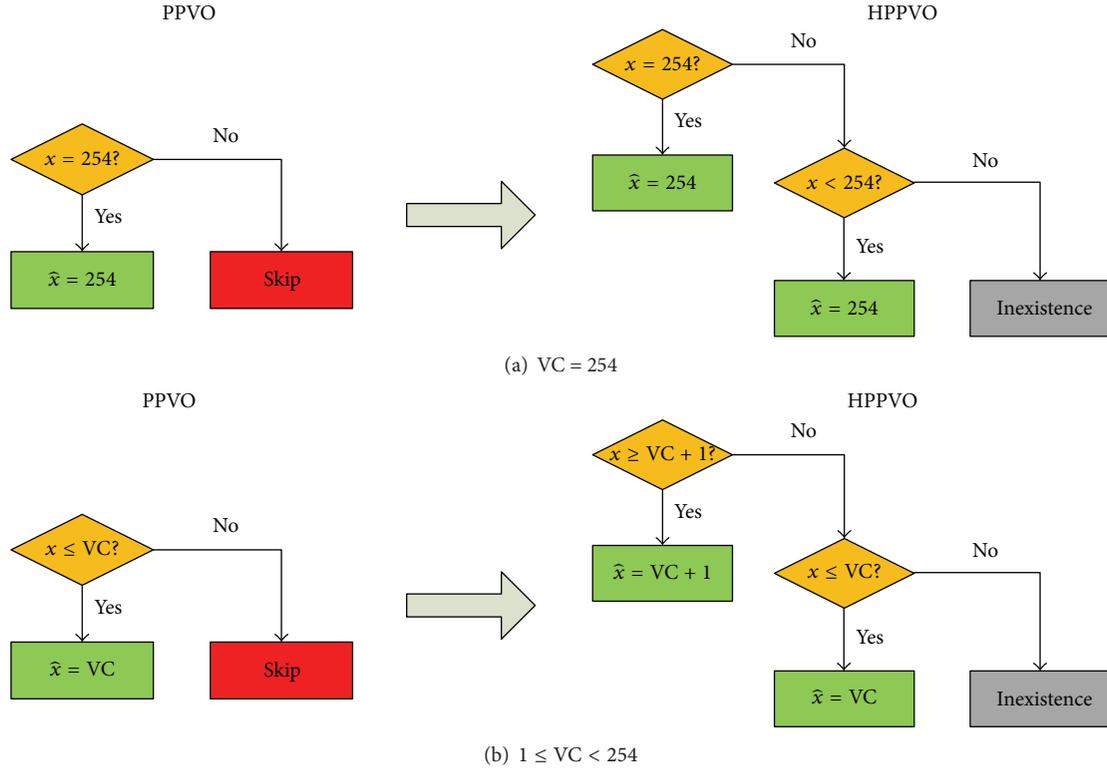


FIGURE 1: The figure puts forward some typical examples and Table 1 illustrates predictions accordingly. Four colors including red, green, yellow, and gray are filled into flowcharts for a straightforward explanation. They, respectively, indicate skip, making prediction, judging conditions, and inexistence.

TABLE 1: Prediction difference of PPVO and the proposed HPPVO.

| Types/prediction | a_1 | a_2 | a_3 | a_4 | a_5 | b_1 | b_2 | b_3 | b_4 | b_5 |
|------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| PPVO/ x | 254 | Skip | Skip | Skip | Skip | Skip | Skip | 230 | 230 | 230 |
| HPPVO/ x | 254 | 254 | 254 | 254 | 254 | 231 | 231 | 230 | 230 | 230 |

performance compared to that of PPVO. Consistent with (2), VC and VC + 1 are probably taken as the prediction value in all possible cases. Conspicuously, this pivotal prediction modification may explore amazing pixel redundancy.

3.1.2. Data Embedding. Accordingly, the derived prediction-error is noted as follows:

$$e = x - \hat{x}. \quad (3)$$

For predicted pixels, prediction-error can be expanded and validated to embed one bit data b or shift for lossless recovery. For nonpredicted pixel, skip it over right away. When context pixel vector possesses unequal values, only those pixels which is equal to their prediction values can be used for embedding. If a pixel is predicted by the maximum of referenced vector, it remains unchanged when data bit is 0 and it is increased by 1 when data bit is 1. However, if a pixel is predicted by the minimum of referenced vector, it will remain unchanged when data bit is 0 and it will be decreased by 1 when data bit is 1. Otherwise, if prediction-error is unequal to zero, the pixel demands of the unit shift.

Thus, when inequality $\max(C) \neq \min(C)$ is satisfied, the original prediction-error e is expanded to be \tilde{e} as follows:

$$\tilde{e} = \begin{cases} e + b & \text{if } \hat{x} = \max(C), e = 0, \\ e + 1 & \text{if } \hat{x} = \max(C), e > 0, \\ e - b & \text{if } \hat{x} = \min(C), e = 0, \\ e - 1 & \text{if } \hat{x} = \min(C), e < 0. \end{cases} \quad (4)$$

Here, b is the to-be-embedded data bit, $b \in \{0, 1\}$.

Different from the previous work in PPVO, when context pixel vectors contain equivalent values, data may be embedded in two situations. First, if the current pixel is predicted by VC + 1, prediction-error zero is expanded. One-bit data 0 or 1 can be embedded and pixel value may remain the same or be increased by 1 at most. It is worth noting that VC + 1 but not VC is the baseline for prediction. So, actual expanded prediction-error amounts to +1. Second, if context pixels are 254, all the possible current pixel can be predicted by 254 and prediction-error zero will be expanded for data embedding. This is contained by the prediction VC. When bit

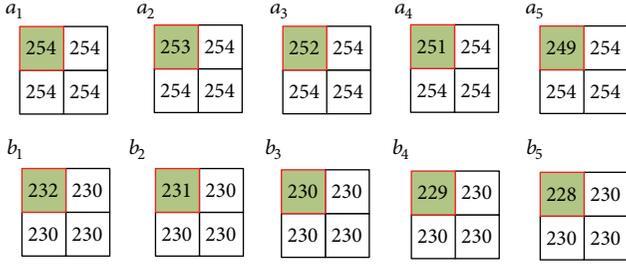


FIGURE 2: The top-left pixel is the current pixel x up for being predicted. Other ones in certain blocks are context pixels. The number of used context pixels is 3. $(a_1) \sim (a_5)$ are mapped from (a) in Figure 1. $(b_1) \sim (b_5)$ are mapped from (b) in Figure 1.

0 is embedded, the pixel remains unchanging and when bit 1 is embedded, the pixel is decreased by 1. Note that though VC is the baseline for judgment, if prediction-error is not equal to 0, the pixel is shifted. When the pixel is greater than predicted value, it is increased by 1 and when pixel is smaller than prediction by 1 or even less, it will be decreased by 1.

When equation $\max(C) = \min(C) = VC$ is established, the initial prediction-error e is expanded to be \tilde{e} as follows:

$$\tilde{e} = \begin{cases} e + b & \text{if } \hat{x} = VC + 1, e = 0, \\ e + 1 & \text{if } \hat{x} = VC + 1, e > 0, \\ e - b & \text{if } \hat{x} = VC, e = 0, \\ e - 1 & \text{if } \hat{x} = VC, e < 0. \end{cases} \quad (5)$$

After embedding, pixel is modified as follows:

$$\tilde{x} = \hat{x} + \tilde{e}. \quad (6)$$

So, in the case of $\max(C) \neq \min(C)$, marked pixels are defined as follows:

$$\tilde{x} = \begin{cases} x + b & \text{if } x = \max(C), \\ x + 1 & \text{if } x > \max(C), \\ x - b & \text{if } x = \min(C), \\ x - 1 & \text{if } x < \min(C). \end{cases} \quad (7)$$

In the case of $\max(C) = \min(C) = VC$, marked pixels are defined as follows:

$$\tilde{x} = \begin{cases} x + b & \text{if } x = VC + 1, \\ x + 1 & \text{if } x > VC + 1, \\ x - b & \text{if } x = VC, \\ x - 1 & \text{if } x < VC. \end{cases} \quad (8)$$

In Figure 3(a), both the current pixel and context pixel are equal to 254 in the first case. Data bit can be embedded on the basis of expanded prediction-error zero. But note that PPVO implements increasing modification $\tilde{x} = x + b$ and HPPVO carries decreasing transformation $\tilde{x} = x - b$. When context

pixels become 253 or other numbers less than 254, the current pixel has to be skipped without prediction in PPVO. However, HPPVO provides a chance for embedding. It takes $VC + 1 = 254$ as the judgment baseline and gets prediction-error zero. After being expanded, this error carries one-bit data. This just makes great difference from PPVO and improves embedding capacity sufficiently. Subsequently, if the current pixel is equal to or less than VC, PPVO and HPPVO execute same arithmetic operations. In Figure 3(b), HPPVO predicts current pixels larger than context pixels. If the current pixel has the value larger by 2 than VC, it will be shifted in HPPVO while being skipped even without prediction in PPVO.

Totally, HPPVO predicts current pixels by two baselines in the case of equivalent context pixels, which is called double prediction mechanism (DPM). This scheme is able to gain predictions not only for pixels equal to or fewer than VC but also for larger ones. Comparatively speaking, PPVO offers a single prediction mechanism (SPM), which cannot predict pixels larger than VC. Thus, more chances stand for prediction in HPPVO rather than in PPVO. This means embedding capacity will be improved by this hybrid predictor in HPPVO.

3.1.3. Error Expansion. To testify the advantage of the new proposed hybrid prediction and embedding method, error expansion is offered some contrast in Figure 4. Integer dotted errors are distributed on coordinate axis with finite extension in both the plus and minus directions. Red arrow denotes error expansion and blue arrow indicates error shift. For (a), nonpositive errors are limited in $[-253, 0]$ existing. If VC is 254, only error 0 is expanded. If VC is not equal to 254, error 0 is expanded and errors less than 0 are shifted in negative-going. For (b), integer errors are in the region of $[-253, 253]$. If prediction value is equal to $VC + 1$, error 0 is expanded and other positive errors are shifted in forward direction. If pixel is predicted by VC, error 0 is expanded and other negative errors are shifted in inverse way. Due to the above, zero error here contains abundant significance. In fact, error 0 produced by $VC + 1$ amounts to error 1 judged by VC. 0^+ and 0^- are used to make these two errors 0 with different meaning distinct.

3.1.4. Bit Extraction and Pixel Recovery. In view of the above-mentioned procedure for hiding data, the embedded data and the host image must be perfectly restored from the marked image. Mathematically, inverse calculation of embedding should be implemented. The same prediction logic as embedding is needed first. All the context pixels for prediction must be the same as the embedded pixels, so searching for pixels must involve an anti-raster-scan order.

To predict the current marked pixel \tilde{x} , prediction-error is noted as follows:

$$\tilde{e} = \tilde{x} - \hat{x}. \quad (9)$$

Referring to (4), in the case of different values in the context pixel vector, the maximum $\max(C)$ or the minimum $\min(C)$ is taken as prediction. In the first case, when error \tilde{e} is zero, one-bit 0 is extracted and the according original

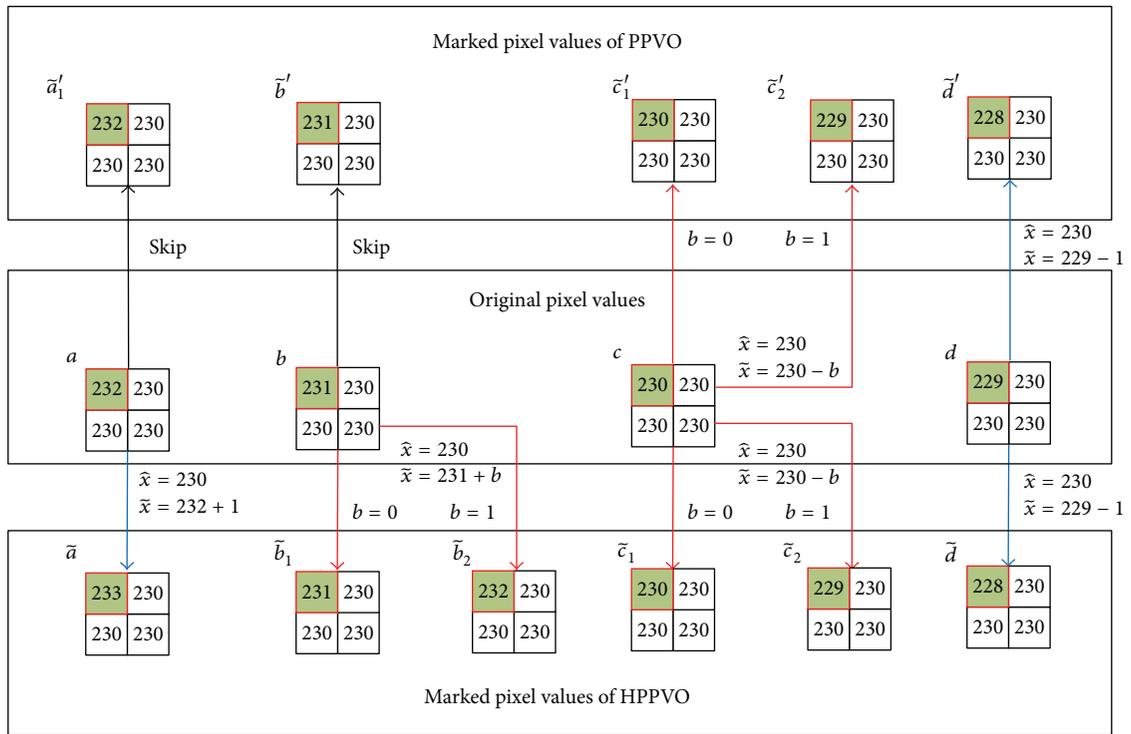
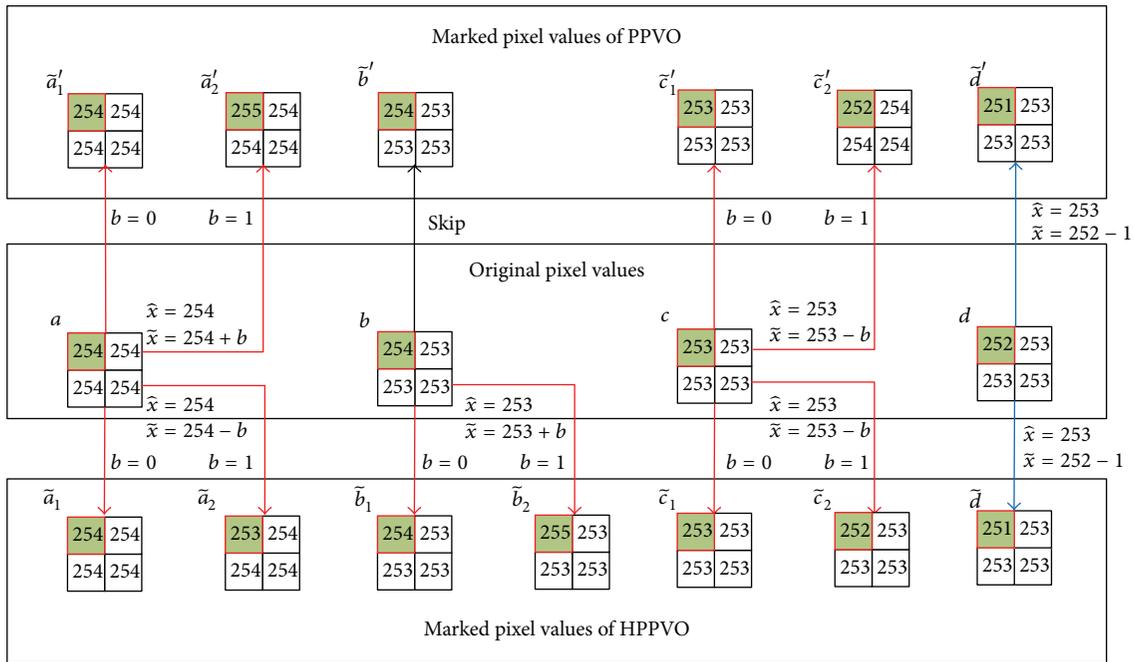


FIGURE 3: PPVO and HPPVO embedding schemes in the case of equivalent context pixels. The top-left pixel is the current pixel to be predicted. There are three context pixels and the pixel values are equal to a constant VC. The red solid lines indicate embedded data bits. The blue solid lines indicate shifts. The black solid lines indicate skipping. Rows indicate pixel transformation.

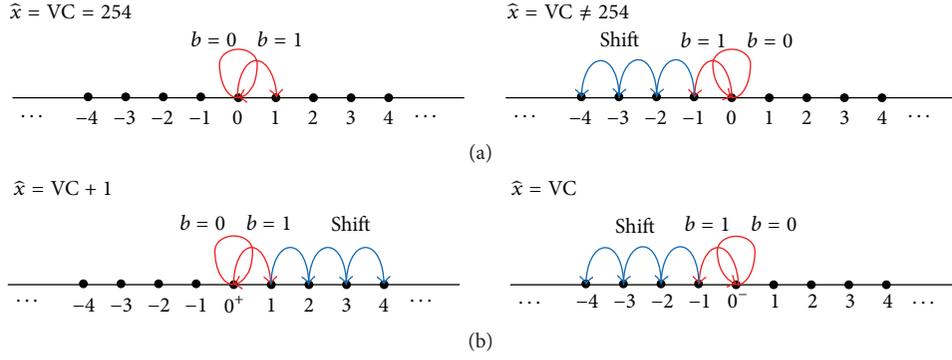


FIGURE 4: Error expansions. Red arrow stands for expanded errors. Blue arrow is shifted. b is the data bit to be embedded. (a) shows two cases in PPVO; (b) stands for the corresponding cases in HPPVO.

pixel is just the same as the current marked pixel. When error \tilde{e} equals one, one-bit 1 is drawn and subtracting 1 from the current pixel makes the original value. When error \tilde{e} is larger than 1, no bit will be extracted, but one unit must be shifted by subtraction. In the second case, if error $\tilde{e} = 0$, the extracted bit $b = 0$ and the pixel holds unchanged. If error $\tilde{e} = -1$, the extracted bit $b = 1$ and the pixel is increased by 1. If error $\tilde{e} < -1$, no bit can be drawn and the pixel is shifted by increasing it by 1. As shown in Section 3.2, only prediction-error 0 is operated. In other cases, no prediction can be obtained and skip this pixel.

Thus, when $\max(C) \neq \min(C)$, error \tilde{e} is decoded as follows:

$$e = \begin{cases} \tilde{e}, b = 0 & \text{if } \hat{x} = \max(C), \tilde{e} = 0, \\ \tilde{e} - 1, b = 1 & \text{if } \hat{x} = \max(C), \tilde{e} = 1, \\ \tilde{e} - 1 & \text{if } \hat{x} = \max(C), \tilde{e} > 1, \\ \tilde{e}, b = 0 & \text{if } \hat{x} = \min(C), \tilde{e} = 0, \\ \tilde{e} + 1, b = 1 & \text{if } \hat{x} = \min(C), \tilde{e} = -1, \\ \tilde{e} + 1 & \text{if } \hat{x} = \min(C), \tilde{e} < -1. \end{cases} \quad (10)$$

Referring to (5), in the condition of all the context pixels equal to a certain number VC , both $VC + 1$ and VC are candidate prediction baselines. Errors 0 and -1 are already decoded for hybrid extraction. When pixel is predicted by $VC + 1$, three cases have to be considered. If error is 0, data bit is 0 and pixel is the same without any modification. If error is 1, $b = 1$ is extracted. Otherwise, error is larger than 1, pixel is shifted, and no bit can be decoded. In the last two cases, pixel will be decreased by 1 for lossless recovery. When pixel is predicted by VC , similar three cases demand for consideration. If error is 0, data bit is 0 and pixel is the same without any modification. If error is -1 , $b = 1$ is extracted. Otherwise, error is less than -1 , pixel is shifted, and no bit can be decoded. For the last two scenarios, pixel will be increased by 1 for image recovery.

When $\max(C) = \min(C) = VC$, error \tilde{e} is restored as follows:

$$e = \begin{cases} \tilde{e}, b = 0 & \text{if } \hat{x} = VC + 1, \tilde{e} = 0, \\ \tilde{e} - 1, b = 1 & \text{if } \hat{x} = VC + 1, \tilde{e} = 1, \\ \tilde{e} - 1 & \text{if } \hat{x} = VC + 1, \tilde{e} > 1, \\ \tilde{e}, b = 0 & \text{if } \hat{x} = VC, \tilde{e} = 0, \\ \tilde{e} + 1, b = 1 & \text{if } \hat{x} = VC, \tilde{e} = -1, \\ \tilde{e} + 1 & \text{if } \hat{x} = VC, \tilde{e} < -1. \end{cases} \quad (11)$$

Recovered pixels are as follows:

$$x = \hat{x} + e. \quad (12)$$

Under the condition of $\max(C) \neq \min(C)$, the current pixel is recovered as follows:

$$x = \begin{cases} \tilde{x}, b = 0 & \text{if } \tilde{x} = \max(C), \\ \tilde{x} - 1, b = 1 & \text{if } \tilde{x} = \max(C) + 1, \\ \tilde{x} - 1, \text{no bit is extracted} & \text{if } \tilde{x} > \max(C) + 1, \\ \tilde{x}, b = 0 & \text{if } \tilde{x} = \min(C), \\ \tilde{x} + 1, b = 1 & \text{if } \tilde{x} = \min(C) - 1, \\ \tilde{x} + 1, \text{no bit is extracted} & \text{if } \tilde{x} < \min(C) - 1. \end{cases} \quad (13)$$

Under the condition of $\max(C) = \min(C) = VC$, the current pixel is recovered as follows:

$$x = \begin{cases} \tilde{x}, b = 0 & \text{if } \tilde{x} = VC + 1, \\ \tilde{x} - 1, b = 1 & \text{if } \tilde{x} = VC + 2, \\ \tilde{x} - 1, \text{no bit is extracted} & \text{if } \tilde{x} > VC + 2, \\ \tilde{x}, b = 0 & \text{if } \tilde{x} = VC, \\ \tilde{x} + 1, b = 1 & \text{if } \tilde{x} = VC - 1, \\ \tilde{x} + 1, \text{no bit is extracted} & \text{if } \tilde{x} < VC - 1. \end{cases} \quad (14)$$

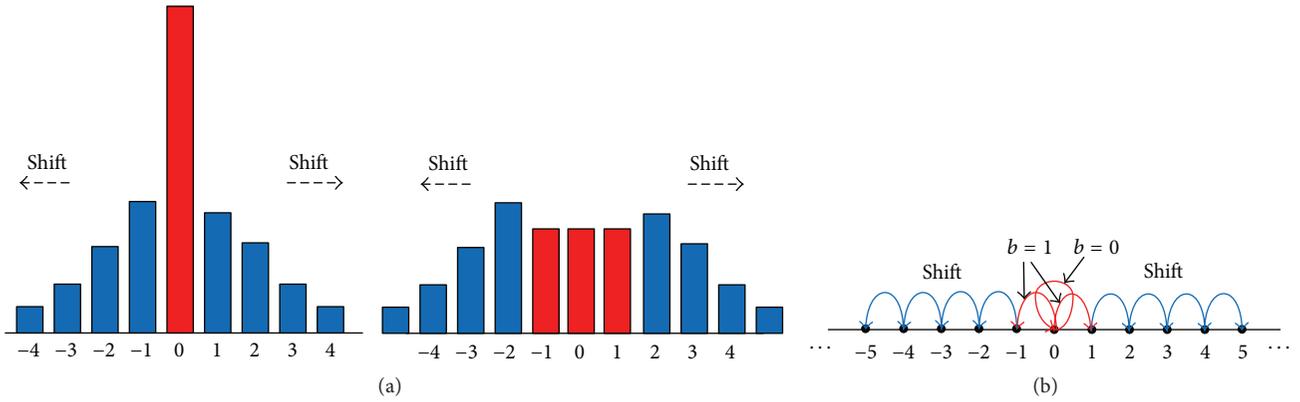


FIGURE 5: Histogram modification mechanism for Lena image without EBS. Red bins stand for expanded errors. Blue bins are shifted. Red arrow stands for expanded errors. Blue arrow is shifted. b is the data bit to be embedded. (a) The left is the PEH before and the right is the according one after embedding. (b) Mapping of bins.

3.2. Expansion Bins Selection. Prediction-error histogram (PEH) is usually to demonstrate the frequencies of prediction-errors for certain image. Suppose that N cover pixels denoted by (x_1, \dots, x_N) are collected for embedding. With (3), prediction-error is calculated by

$$e_i = x_i - \hat{x}_i, \quad (15)$$

where $i \in [1, N]$. PEH can be derived as

$$h(e) = \# \{1 \leq i \leq N : e_i = e\}, \quad \forall e \in \mathbb{Z}, \quad (16)$$

where $\#$ means a cardinal number set. Consider here a PEH for Lena image illustrated by Figure 5, which is generated from the above proposed HPPVO.

Figure 5 describes prediction-error bins in common. With (4) and (5), only error 0 bin is expanded and other bins are shifted to make vacancies for lossless reversibility. For the left figure in (a), 0 bin forms a rather sharp peak before embedding. In line with (16), note histogram bin $h(e_i)$ for error e_i in (e_1, \dots, e_N) . For the right figure in (a), it is partitioned into three parts after embedding. One part will remain in original post for $b = 0$ and the other two parts will shift towards the right and left for $b = 1$, which is caused by (3), (4), and (5). Positive error bins pass on their right locations and negative bins move to adjacent left places. (b) shows error expansion using red arrow and shift by blue arrow. On this occasion, the expected embedding distortion in l^2 -norm can be computed by

$$\begin{aligned} E_{(0,0)}^d &= E \left(\|\tilde{I} - I\|_2^2 \right) = \sum_{i=1}^N E \left((\tilde{x}_i - x_i)^2 \right) \\ &= \frac{2}{3}h(0) + \sum_{\substack{e \notin \{0\}, \\ e \in \{-253, \dots, 253\}}} h(e) = N - \frac{h(0)}{3}, \end{aligned} \quad (17)$$

where I and \tilde{I} denote cover and marked images, respectively. However, one can select the other bin or bins for expansion rather than 0.

From the above, data bit can be embedded for both positive and negative prediction-error either for error 0. Figures 6, 7, and 8 display cases for several distinct coupled expansion bins (e_l, e_r) . e_l and e_r indicate one side of a pair of expansion bins satisfying $e_l \in [-253, 0]$ and $e_r \in [0, 253]$. Therefore, histogram modification mechanism for Lena image without EBS in Figure 5 can be considered to be a typical case $(0, 0)$. When $(e_l, e_r) = (-1, 0)$ illustrated in Figure 6, -1 bin can be employed to carry data bits. Parts of error -1 are unchanged for embedding $b = 0$ and the others are shifted to error -2 for carrying $b = 1$. In this situation, the likely embedding distortion can be formulated as

$$\begin{aligned} E_{(-1,0)}^d &= E \left(\|\tilde{I} - I\|_2^2 \right) = \sum_{i=1}^N E \left((\tilde{x}_i - x_i)^2 \right) \\ &= \frac{2}{3}h(0) + \frac{1}{2}h(-1) + \sum_{\substack{e \notin \{-1,0\}, \\ e \in \{-253, \dots, 253\}}} h(e) \quad (18) \\ &= N - \frac{h(0)}{3} - \frac{1}{2}h(-1). \end{aligned}$$

When $(e_l, e_r) = (0, 1)$ showed in Figure 7, 1 bin can be employed to carry data bits. Parts of error 1 are unmodified for embedding $b = 0$ and the others are moved to the place of error 2 for carrying $b = 1$. This declares more embedding capacity than that in Figure 5. In this case, the conceivable distortion will be counted as

$$\begin{aligned} E_{(0,1)}^d &= E \left(\|\tilde{I} - I\|_2^2 \right) = \sum_{i=1}^N E \left((\tilde{x}_i - x_i)^2 \right) \\ &= \frac{2}{3}h(0) + \frac{1}{2}h(1) + \sum_{\substack{e \notin \{0,1\}, \\ e \in \{-253, \dots, 253\}}} h(e) \quad (19) \\ &= N - \frac{h(0)}{3} - \frac{1}{2}h(1). \end{aligned}$$

When $(e_l, e_r) = (-1, 1)$ showed in Figure 8, both -1 and 1 bins can be used to carry data bits. Parts of them are

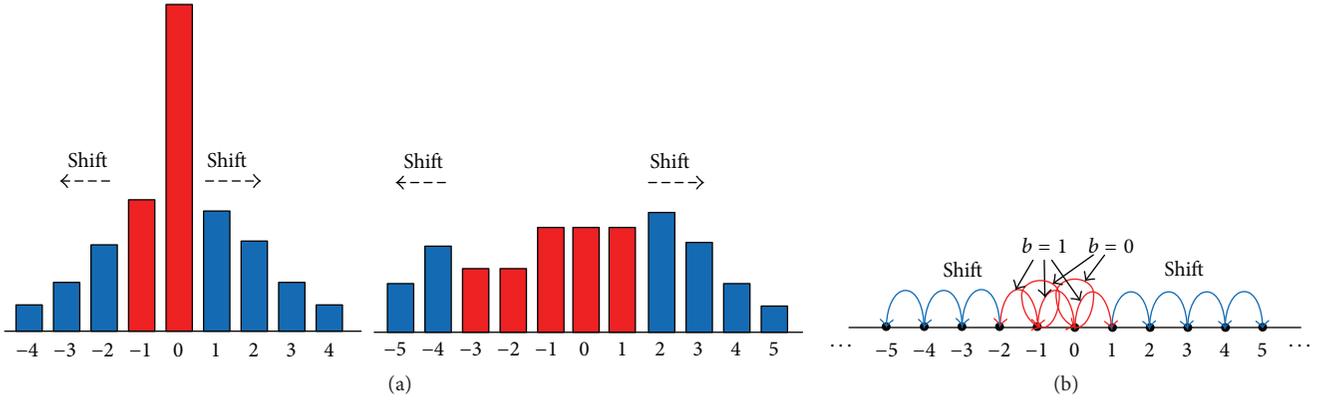


FIGURE 6: Histogram modification mechanism for Lena image with coupled expansion bins $(-1, 0)$. (a) The left is the PEH before and the right is the according one after embedding. (b) Mapping of bins.

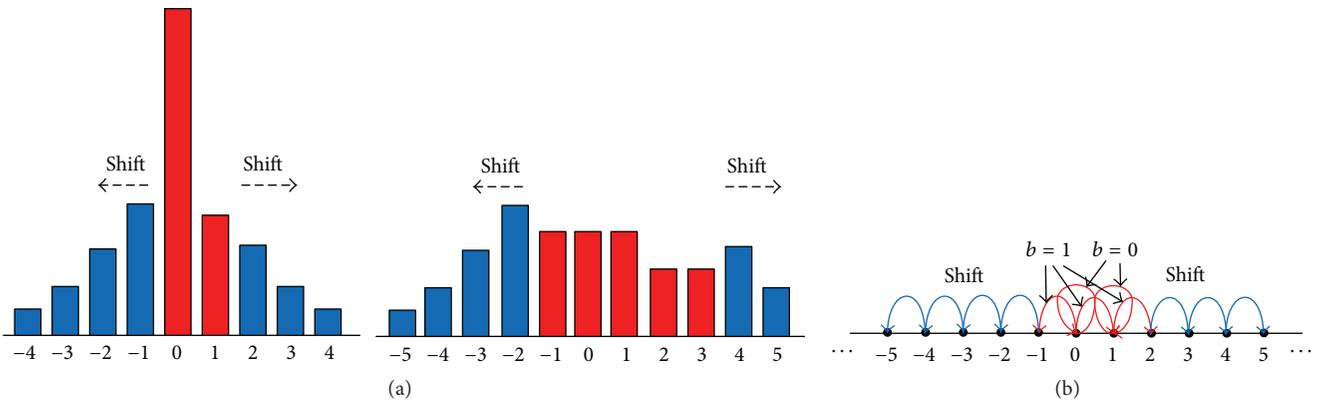


FIGURE 7: Histogram modification mechanism for Lena image with coupled expansion bins $(0, 1)$. (a) The left is the PEH before and the right is the according one after embedding. (b) Mapping of bins.

unmodified for embedding $b = 0$. But -1 has to move to left and 1 must shift towards right for adjacent post to embed a bit $b = 1$. It is a remarkable fact that 0 bin occupying overwhelming pixels is skipped. This will improve embedding performance sufficiently for the following expression:

$$\begin{aligned}
 E_{(-1,1)}^d &= E \left(\|\tilde{I} - I\|_2^2 \right) = \sum_{i=1}^N E \left((\tilde{x}_i - x_i)^2 \right) \\
 &= \frac{1}{2} (h(-1) + h(1)) + \sum_{\substack{e \notin \{-1,0,1\}, \\ e \in \{-253, \dots, 253\}}} h(e) \quad (20) \\
 &= N - h(0) - \frac{1}{2}h(-1) - \frac{1}{2}h(1).
 \end{aligned}$$

On the whole, EBS will improve embedding capacity shown in Figures 5–8. In addition, taking a comparison with (17) by (18), (19), and (20), the inequation denoted by (21) comes into being. So, it can be concluded that expansion bins selecting strategy contributes to bringing

down embedding distortion. Thus, it is probable for better embedding performance using EBS that

$$\begin{aligned}
 N - \frac{h(0)}{3} &\leq \min \left(\left(N - \frac{h(0)}{3} - \frac{1}{2}h(-1) \right), \right. \\
 &\quad \left. \left(N - \frac{h(0)}{3} - \frac{1}{2}h(1) \right) \right) \\
 &\leq \max \left(\left(N - \frac{h(0)}{3} - \frac{1}{2}h(-1) \right), \right. \\
 &\quad \left. \left(N - \frac{h(0)}{3} - \frac{1}{2}h(1) \right) \right) \leq N - h(0) - \frac{1}{2}h(-1) \\
 &\quad - \frac{1}{2}h(1).
 \end{aligned} \quad (21)$$

3.3. Field-Biased Context Pixel Selection (FCPS). According to different functions, pixels can be categorized into two types. One is to-be-predicted type and the other one is auxiliary prediction type. The latter pixels are also known as context pixels. For simplicity, we call the to-be-predicted pixel the current pixel. Weighted example introduced in [27] will be analyzed further. To be more intuitive, the different

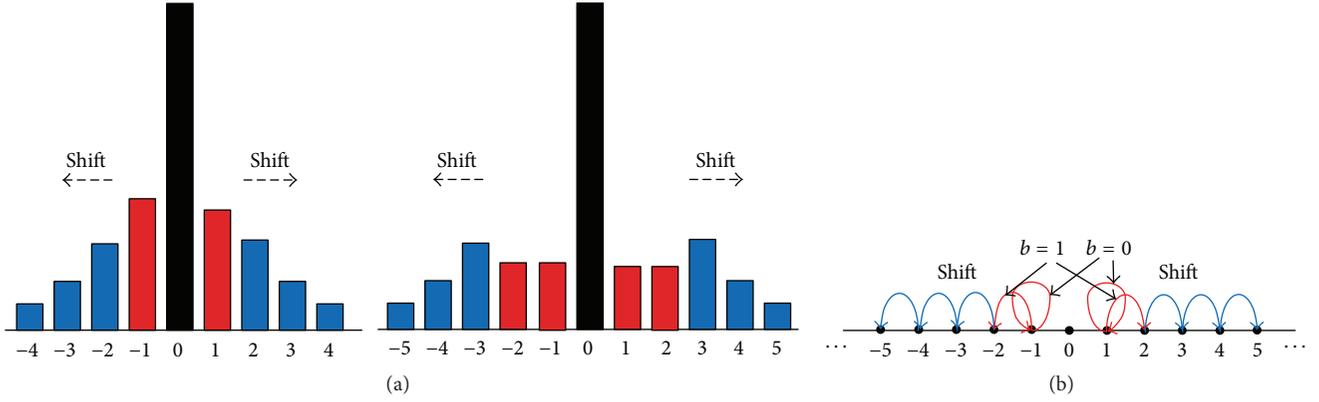


FIGURE 8: Histogram modification mechanism for Lena image with coupled expansion bins $(-1, 1)$. Red bins stand for expanded errors. Blue bins are shifted. Black bins are skipped. (a) The left is the PEH before and the right is the according one after embedding. (b) Mapping of bins.

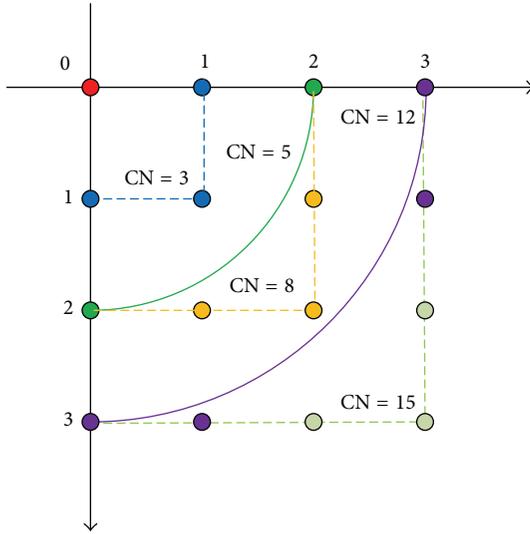


FIGURE 9: Equalizing distribution of context pixels in PPVO (CN = 3, 5, 8, 12, and 15).

distribution of context pixels is given which is illustrated in Figure 9 for different pixels number denoted by CN. When CN = 3, 8, and 15, context pixels locate in the inner and marginal regions of a square, laid at the top-left vertex by the current pixel and lengthened by pixel intervals of 1, 2, and 3 times unit pixel intervals, respectively. While CN = 5 or 12, context pixels are distributed in the inner and marginal regions of a right angle sector, whose radii are 2 or 3 times unit pixel intervals. In the sight of pixel locations, context pixels dot certain organized areas, such as squares and right-angle sectors. So, this is called well-balanced distribution or equalizing distribution.

Actually, the number and distribution of context pixels represent some correlation information about the current pixel. It can be interpreted as gray value variation of image pixels. Difference $e_i = x_i - x_o$ may be employed to represents this changing process, where x_i is the current pixel and x_o is a selected context pixel. This difference value stands for

the similarity degree of two pixels. If e_i tends to zero still more, the according two pixels are more similar. Otherwise, similar degree declines. The larger CN is, the more the specifically rounded pixels evolve.

In terms of pixel similarity, distance potential can be defined as follows.

Definition. Based on a certain baseline pixel, distance potential of another pixel is measured by

$$E = \bar{\varepsilon} \sqrt{k_h^2 + k_v^2} \cdot d_u, \quad (22)$$

where ε is pixel value gradient, d_u indicates the unit horizontal or vertical pixel distance, and k_h, k_v are unit distance times. Assuming that image resolution is $W \times H$, a set is so denoted as $K = \{1, 2, \dots, \max(W, H)\}$. Obviously, $k_h \in K_h, k_v \in K_v$, and $K_h = K_v = K$.

Inspecting the right previous example, the current pixel is the baseline pixel. For a specified CN, here is $\forall k_h \in K_h, \exists k_v \in K_v$ satisfying $k_h = k_v$. Of course, $\max(k_h) = \max(k_v)$ and $\min(k_h) = \min(k_v)$. Considering different CN, when CN = 3, $\vec{k}_h = \vec{k}_v = [1]^T$. When CN = 5 and 8, $\vec{k}_h = \vec{k}_v = [1 \ 2]^T$. When CN = 12 and 15, $\vec{k}_h = \vec{k}_v = [1 \ 2 \ 3]^T$. Thus, context pixels not only depict the similarity between surrounding pixels versus the current pixel, but also demonstrate according distance potential for a certain similar degree.

Comparing with equalizing schemes, field-biased context pixels, shown in Figure 10, have different distance potential. In other words, $\forall k_h \in K_h, \exists k_v \in K_v$ is not always tenable. As far as different CN, k_h and k_v take distinct rules. When CN = 5, $\vec{k}_h = [1 \ 2]^T$ and $\vec{k}_v = [1]^T$. When CN = 7, $\vec{k}_h = [1 \ 2 \ 3]^T$ and $\vec{k}_v = [1]^T$. When CN = 11, $\vec{k}_h = [1 \ 2 \ 3]^T$ and $\vec{k}_v = [1 \ 2]^T$. These six aforementioned schemes discriminatively compress and amplify the variation region of similarity between the current pixel and context pixels in the two dimensions of horizontal and vertical orientation to some extent. These pixels associate some zonal features and therefore we called them field-biased context pixels. From distribution, field-biased context pixels collected difference of horizontal or vertical pixel transformation, which is conducive for describing

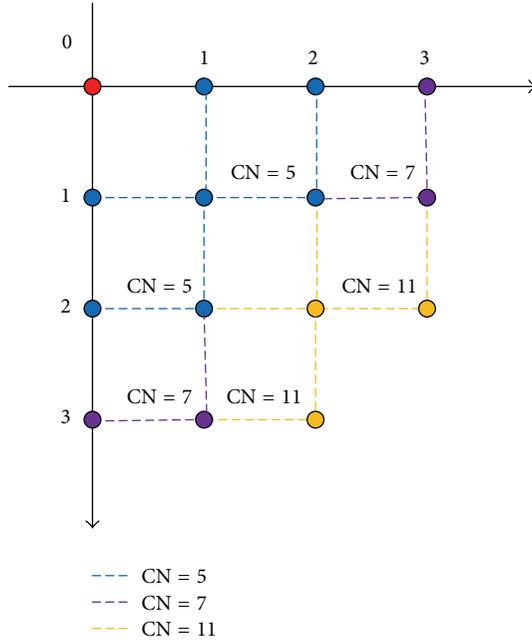


FIGURE 10: Optional field-biased context pixels (CN = 5, 7, and 11).

the image gray value changes in different directions. After introducing offset type context pixels, the current pixel tends to be able to predict more accurately than only equalizing schemes.

4. Experimental Results

Considering the embedding capacity and the peak signal-to-noise ratio (PSNR) value, the performance of the proposed HPPVO is tested with four previous works in [10, 24, 27, 33]. Standard formatted images of eight-bit gray-scale pictures in the size of 512×512 are selected from the SIPI image database. In Figure 11, Lena, Airplane (F16), Baboon, Barbara, Boat, and Peppers meet this demand. Certainly, random bit streams make the payloads for objective test.

4.1. PSNR Performance. Regarding PSNR value, HPPVO outperforms other four involved schemes in most cases, which is shown in Figure 12. Two methods in two previous works in [10, 33] were both found to be capable of embedding large amounts of data. However, PVO and its evolution methods provided much higher PSNR values. PPVO had considerable superiority in both embedding capacity and fidelity relative to three algorithms given above. In all the tested images, HPPVO increases the embedding capacity without decreasing the PSNR value regardless of small or large payload. Especially in smooth and very smooth images, Lena, Barbara, and Airplane (F16) show relatively obvious improvements in moderate and large payload sizes.

4.2. Embedding Capacity Performance. In terms of embedding capacity, Table 2 shows the results in different context pixels. Generally speaking, HPPVO obtains higher EC than

PPVO in most cases. For a certain image, context pixel number has effect on capacity improvement. As shown in Figure 13, Lena is improved in embedding capacity by 446 bits, 15 bits, 1 bit, 4017 bits, and 0 bits and F16 receives capacity gains by 2032 bits, 291 bits, 26 bits, 3325 bits, and 0 bits when the number of context pixels was 3, 5, 8, 12, and 15. This indicates that more data can be embedded into images with HPPVO than that with conventional PPVO. Yet, different CN causes different EC improvement. For more detailed comparison, the same and the adjacent numbers of context pixels are taken together. Here, 3, 5, 8, 12, and 15 are defined as central points. Accordingly, comparisons are drawn by maximum capacity of C_{5h} and C_{5v} with that of C_5 , maximum capacity of C_{7h} and C_{7v} with that of C_8 , and maximum capacity of C_{11h} and C_{11v} with that of C_{12} . Blue marked lines and circles in Figure 13 show that Lena increases by 446 bits, 1991 bits, 1974 bit, 6004 bits, and 0 bits, and F16 image gains improvements of 2032 bits, 3332 bits, 3320 bits, 11948 bits, and 3 bits' capacity. This demonstrates that the proposed hybrid works strikingly with similar context numbers. It is a remarkable fact that when CN is near to 12, prominent improvement can be achieved. And so, in Figure 14, this item will help us to analyze how notable results will be obtained from images with different complexity.

We define the improvement ratio by

$$r_{IEC} = \frac{EC_h - EC_c}{EC_c}, \quad (23)$$

where the subscript h indicates the new proposed hybrid predictor and c denotes the conventional predictor. Based on Table 2, we drafted Figure 14. Clearly, the improved EC ratio to CN = 12 is outstanding among other cases. On this occasion, all images appear well. Lena, Barbara, Boat, and Peppers obtain 0.33, 0.25, 0.26, and 0.26 improvements in common complexity. F16, the smoothest one, gets the maximum ratio value of 0.44. Even the most rough image, Baboon, takes the minimum 0.18. These splendid performances make good annotation for EC improvement of the proposed HPPVO.

From above, some conclusions can be drawn as follows:

- (a) The number of context pixels has an effect on capacity increases. Moderate and large number, here CN = 12, obtain the most prominent performance. This is because nearly half of predictable opportunities from the two-way hybrid predictor will be appended for pixel vacancies. Simultaneously, capacity can be improved for neighboring pixels in field-biased schemes to a greater extent than in equalizing schemes.
- (b) Image complexity takes influence on embedding capacity. Smooth images receive higher capacity more than that of rough ones. This mainly relies on the above simulated examples, F16 and Lena, the most and the second-rate smooth images, acquire the best and the suboptimal capacity improvement, respectively.
- (c) Texture feature produces distinct capacity gains. The texture is more visible, the better enhancement in

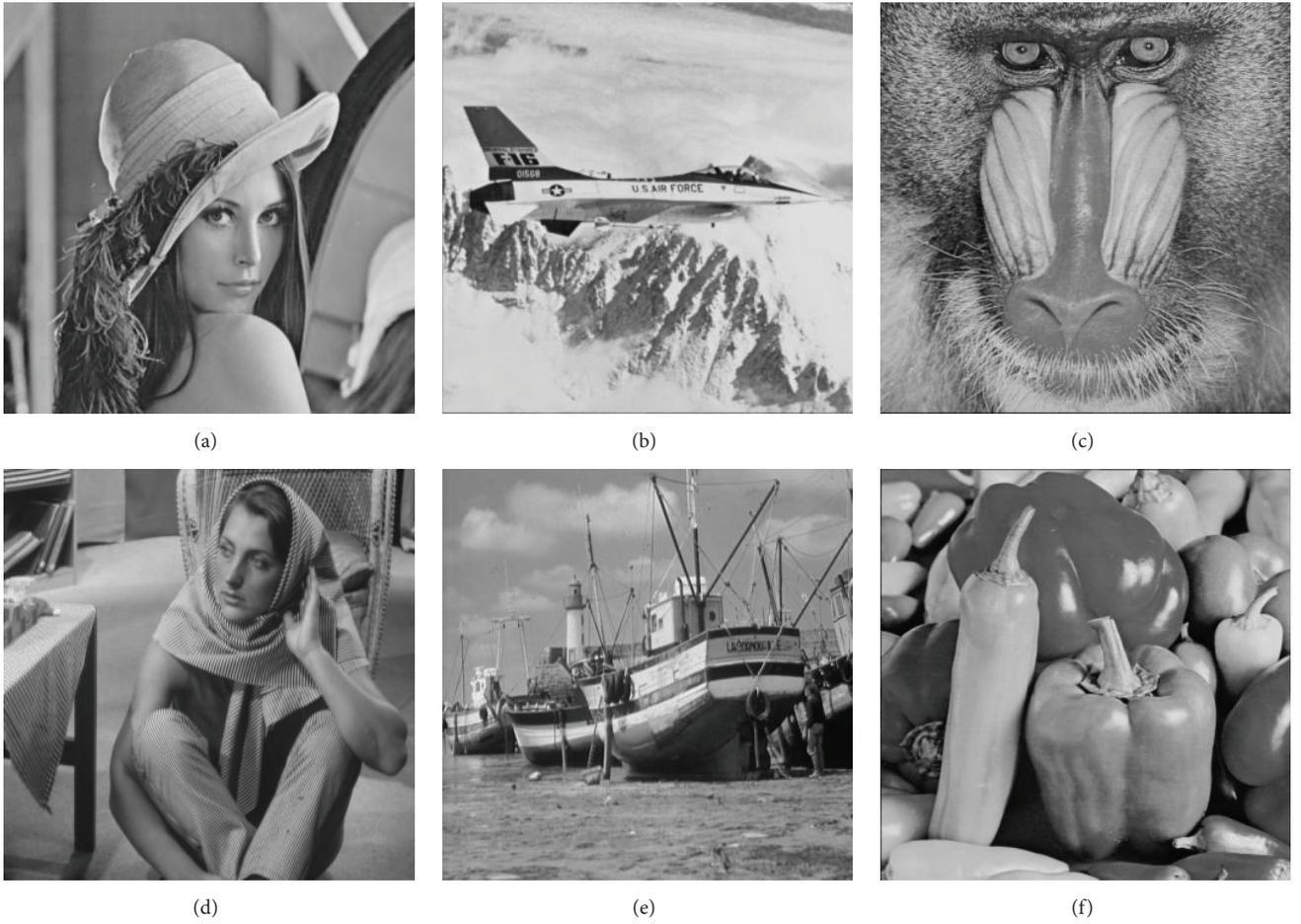


FIGURE 11: Test images of SIPI image data set. (a) Lena. (b) Airplane (F16). (c) Baboon. (d) Barbara. (e) Boat. (f) Peppers.

TABLE 2: Comparisons of embedding capacity (/bit) in terms of different context pixels between PPVO and HPPVO.

| Image | C_3 | C_5 C_{5h}/C_{5v} | C_8 C_{7h}/C_{7v} | C_{12} C_{11h}/C_{11v} | C_{15} |
|------------------------|----------------------|--|--|--|----------------------|
| Lena | 44,118/44,564 | 34,210/34,226 35,825/36,201 | 28,561/28,562 30,535/30,452 | 18,195/22,212 24,199/23,806 | 20,175/20,175 |
| Baboon | 13,548/13,576 | 10,361/10,361 10,830/10,630 | 8,367/8,367 9,172/8,876 | 5,995/6,532 7,047/6,958 | 5,779/5,779 |
| Barbara | 34,930/35,266 | 27,013/27,024 28,416/29,056 | 23,127/23,127 24,480/25,174 | 16,036/18,429 19,873/19,982 | 17,147/17,147 |
| Boat | 29,194/29,369 | 22,930/22,937 24,005/23,741 | 18,946/18,946 20,151/20,096 | 12,622/14,575 15,827/15,891 | 13,126/13,126 |
| Peppers | 33,439/33,628 | 26,341/26,344 27,311/27,368 | 21,524/21,524 23,267/23,265 | 14,469/16,865 18,226/18,249 | 15,435/15,435 |
| F16 | 66,923/68,955 | 53,372/53,663 56,643/56,704 | 45,455/45,518 48,319/48,775 | 26,888/35,540 38,170/38,836 | 32,411/32,414 |
| Cluster Average | 37,025/37,560 | 29,021/29,093 30,505/30,617 | 24,330/24,356 25,987/26,106 | 15,701/19,026 20,557/20,620 | 17,346/17,346 |

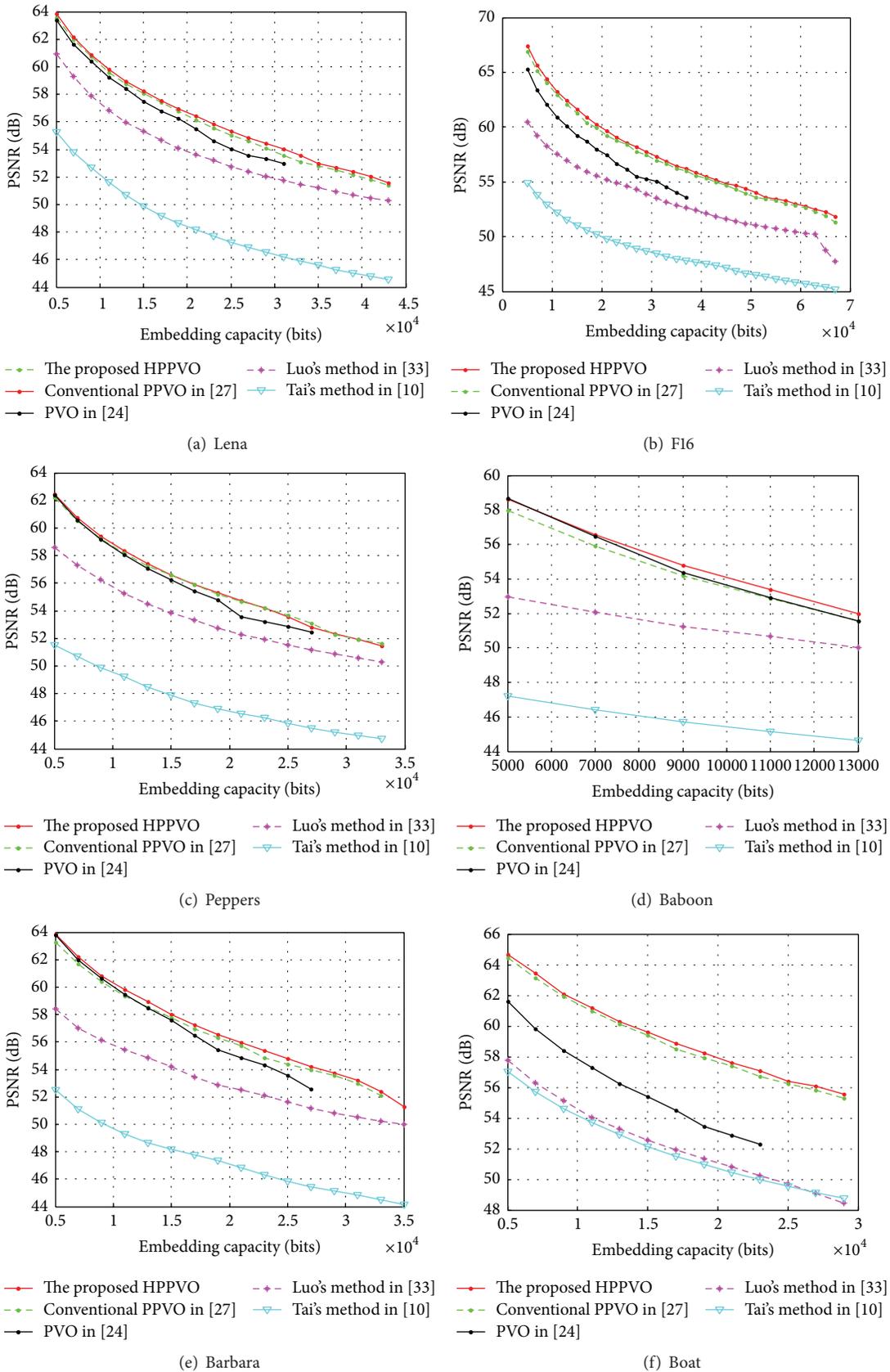


FIGURE 12: The performance comparison of proposed scheme and four state-of-the-art methods in [10, 24, 27, 33]. All the test images are from the SIPI image data set.

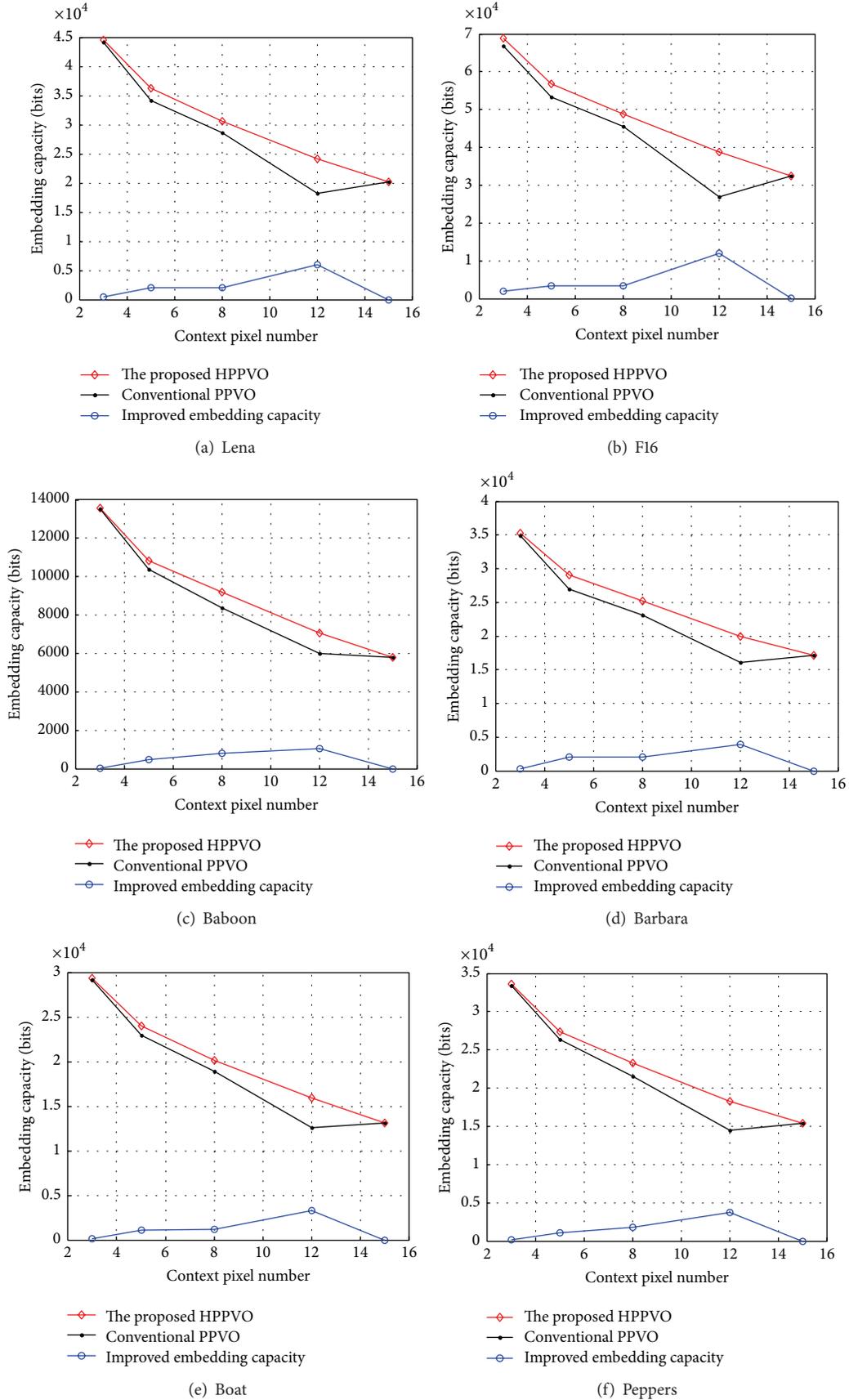


FIGURE 13: Improved capacity comparison for different numbers of context pixels. All the test images are from SIPI image data set.

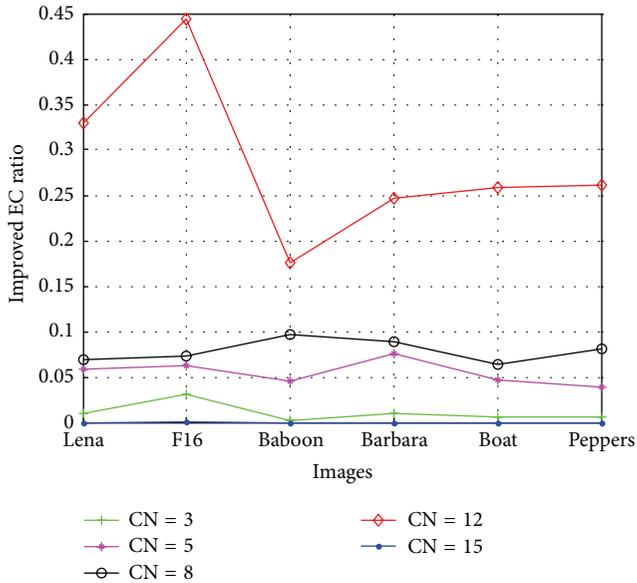


FIGURE 14: Comparison of improved EC ratios to images for different CN.

field-biased context pixel selections. This is because of the variation describing for texture pixels. Pixels in the zonal massif region in F16 image and the diagonal headwear area in Lena image contribute vast expandable prediction-errors.

Totally, HPPVO is more suitable for marked images claiming for high fidelity and large embedding capacity.

5. Conclusion

HPPVO, which is designed to create large embedding capacity without reducing the PSNR value, combines hybrid prediction and field-biased context pixel selections. New embedding logic enhances embedding data, and new context pixel scheme is found to improve PSNR values. Experimental results show that HPPVO takes outstanding advantages in smooth and very smooth images. This suggests that field-biased context pixel schemes merit further research.

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. 61302106 and no. 61201393), the Fundamental Research Funds for the Central Universities (no. 2014MS105 and no. 13MS66), and Hebei Province Natural Science Foundation of China, Youth Science Fund (no. E2013502267). Here, the authors are obliged to the above foundations, their responsible persons, and all their lab associates.

References

- [1] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673–1687, 1997.
- [2] C. I. Podilchuk and E. J. Delp, "Digital watermarking: algorithm and application," *IEEE Signal Processing Magazine*, vol. 18, no. 4, pp. 33–46, 2001.
- [3] G. C. Langelaar, I. Setyawan, and R. L. Lagendijk, "Watermarking digital image and video data," *IEEE Signal Processing Magazine*, vol. 17, no. 5, pp. 20–46, 2000.
- [4] S. Y. Chen, H. Tong, Z. Wang, S. Liu, M. Li, and B. Zhang, "Improved generalized belief propagation for vision processing," *Mathematical Problems in Engineering*, vol. 2011, Article ID 416963, 12 pages, 2011.
- [5] M. Barni, F. Bartolini, and A. Piva, "Improved wavelet-based watermarking through pixel-wise masking," *IEEE Transactions on Image Processing*, vol. 10, no. 5, pp. 783–791, 2001.
- [6] S. Y. Chen, J. Zhang, Q. Guan, and S. Liu, "Detection and amendment of shape distortions based on moment invariants for active shape models," *IET Image Processing*, vol. 5, no. 3, pp. 273–285, 2011.
- [7] H. Liu, S. Y. Chen, and N. Kubota, "Guest editorial special section on intelligent video systems and analytics," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, p. 90, 2012.
- [8] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.
- [9] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147–1156, 2004.
- [10] W.-L. Tai, C.-M. Yeh, and C.-C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 906–910, 2009.
- [11] Z. C. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.
- [12] Y.-S. Juang, L.-T. Ko, J.-E. Chen, Y.-S. Shieh, T.-Y. Sung, and H. C. Hsin, "Histogram modification and wavelet transform for high performance watermarking," *Mathematical Problems in Engineering*, vol. 2012, Article ID 164869, 14 pages, 2012.
- [13] Y.-Y. Tsai, D.-S. Tsai, and C.-L. Liu, "Reversible data hiding scheme based on neighboring pixel differences," *Digital Signal Processing*, vol. 23, no. 3, pp. 919–927, 2013.
- [14] X. L. Li, B. Li, B. Yang, and T. Y. Zeng, "General framework to histogram-shifting-based reversible data hiding," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2181–2191, 2013.
- [15] L.-T. Ko, J.-E. Chen, H.-C. Hsin, Y.-S. Shieh, and T.-Y. Sung, "Haar-wavelet-based just noticeable distortion model for transparent watermark," *Mathematical Problems in Engineering*, vol. 2012, Article ID 635738, 14 pages, 2012.
- [16] S. Lee, C. D. Yoo, and T. Kalker, "Reversible image watermarking based on integer-to-integer wavelet transform," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 321–330, 2007.
- [17] X. Wang, X.-L. Li, B. Yang, and Z.-M. Guo, "Efficient generalized integer transform for reversible watermarking," *IEEE Signal Processing Letters*, vol. 17, no. 6, pp. 567–570, 2010.
- [18] L.-T. Ko, J.-E. Chen, Y.-S. Shieh, M. Scialia, and T.-Y. Sung, "A novel fractional-discrete-cosine-transform-based reversible

- watermarking for healthcare information management systems,” *Mathematical Problems in Engineering*, vol. 2012, Article ID 757018, 17 pages, 2012.
- [19] D. Coltuc, “Low distortion transform for reversible watermarking,” *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 412–417, 2012.
- [20] S.-K. Lee, Y.-H. Suh, and Y.-S. Ho, “Reversible image authentication based on watermarking,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '06)*, vol. 1–5, pp. 1321–1324, IEEE, Toronto, Canada, July 2006.
- [21] L. Kamstra and H. J. Heijmans, “Reversible data embedding into images using wavelet techniques and sorting,” *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2082–2090, 2005.
- [22] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, “Reversible watermarking algorithm using sorting and prediction,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 989–999, 2009.
- [23] D. M. Thodi and J. J. Rodriguez, “Expansion embedding techniques for reversible watermarking,” *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 721–730, 2007.
- [24] X. L. Li, J. Li, B. Li, and B. Yang, “High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion,” *Signal Processing*, vol. 93, no. 1, pp. 198–205, 2013.
- [25] B. Ou, X. L. Li, Y. Zhao, and R. R. Ni, “Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion,” *Signal Processing: Image Communication*, vol. 29, no. 7, pp. 760–772, 2014.
- [26] F. Peng, X. L. Li, and B. Yang, “Improved PVO-based reversible data hiding,” *Digital Signal Processing*, vol. 25, no. 1, pp. 255–265, 2014.
- [27] X. Qu and H. J. Kim, “Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding,” *Signal Processing*, vol. 111, pp. 249–260, 2015.
- [28] X. Li, B. Yang, and T. Y. Zeng, “Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection,” *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3524–3533, 2011.
- [29] C. Wang, X. Li, and B. Yang, “Efficient reversible image watermarking by using dynamical prediction-error expansion,” in *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP '10)*, pp. 3673–3676, September 2010.
- [30] H. J. Hwang, H. J. Kim, V. Sachnev, and S. H. Joo, “Reversible watermarking method using optimal histogram pair shifting based on prediction and sorting,” *KSII Transactions on Internet and Information Systems*, vol. 4, no. 4, pp. 655–670, 2010.
- [31] H.-T. Wu and J. Huang, “Reversible image watermarking on prediction errors by efficient histogram modification,” *Signal Processing*, vol. 92, no. 12, pp. 3000–3009, 2012.
- [32] G. Xuan, X. Tong, J. Teng, X. Zhang, and Y. Q. Shi, “Optimal histogram-pair and prediction-error based image reversible data hiding,” in *Digital Forensics and Watermarking: 11th International Workshop, IWDW 2012, Shanghai, China, October 31–November 3, 2012, Revised Selected Papers*, vol. 7809 of *Lecture Notes in Computer Science*, pp. 368–383, Springer, Berlin, Germany, 2012.
- [33] L. X. Luo, Z. Y. Chen, M. Chen, X. Zeng, and Z. Xiong, “Reversible image watermarking using interpolation technique,” *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 187–193, 2010.

Research Article

Normal Inverse Gaussian Model-Based Image Denoising in the NSCT Domain

Jian Jia,¹ Yongxin Zhang,^{2,3} Li Chen,² and Zhihua Zhao²

¹School of Mathematics, Northwest University, Xi'an 710127, China

²School of Information Science and Technology, Northwest University, Xi'an 710127, China

³Luoyang Normal University, Luoyang 471022, China

Correspondence should be addressed to Jian Jia; jiajian@nwu.edu.cn

Received 15 October 2015; Revised 8 December 2015; Accepted 9 December 2015

Academic Editor: Daniel Zaldivar

Copyright © 2015 Jian Jia 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 objective of image denoising is to retain useful details while removing as much noise as possible to recover an original image from its noisy version. This paper proposes a novel normal inverse Gaussian (NIG) model-based method that uses a Bayesian estimator to carry out image denoising in the nonsampled contourlet transform (NSCT) domain. In the proposed method, the NIG model is first used to describe the distributions of the image transform coefficients of each subband in the NSCT domain. Then, the corresponding threshold function is derived from the model using Bayesian maximum *a posteriori* probability estimation theory. Finally, optimal linear interpolation thresholding algorithm (OLI-Shrink) is employed to guarantee a gentler thresholding effect. The results of comparative experiments conducted indicate that the denoising performance of our proposed method in terms of peak signal-to-noise ratio is superior to that of several state-of-the-art methods, including BLS-GSM, K-SVD, BivShrink, and BM3D. Further, the proposed method achieves structural similarity (SSIM) index values that are comparable to those of the block-matching 3D transformation (BM3D) method.

1. Introduction

The objective of image denoising, a classical but still very active area in image processing, is to retain useful details while removing as much noise as possible to recover the original image from a noisy image [1]. Following the development of the fast wavelet decomposition method by Mallat [2], wavelet transform has been used extensively in various fields. Its use rapidly extended from mathematics and signal processing to other fields, such as image denoising, an important image processing technique that has developed rapidly with the introduction of wavelets.

The threshold method, developed by Donoho [3] in 1995, provides a viable treatment option for the wavelet coefficients of nonlinear processing and, consequently, significantly advanced the field of image denoising. Other feasible wavelet-based approaches have also been developed. Hard and soft thresholds emerged first, followed by Bayesian threshold [4, 5] methods that significantly enhanced the

denoised results. Recently, various intrascale correlation coefficient threshold methods and a coefficient threshold method under high-dimensional space [6] have also been developed. Through continuous improvement of the threshold method, wavelet denoising methods have achieved increasingly better denoised results. These methods are supported by the underlying wavelet theory. Further, the denoising effect of the wavelet continues to improve with active research being carried out on the scaling relation in the transform domain.

In general, there are two key questions associated with image denoising based on wavelet domain statistical models: (1) how to choose the prior distribution model of the wavelet coefficients and (2) how to determine the appropriate denoising algorithm with respect to the distribution model. The most important topic in wavelet denoising is the wavelet coefficients statistical model. The objective of the research being conducted in this area is to make an accurate model of the wavelet coefficients, which are non-Gaussian and related to each other. Chang et al. [7] modelled the prior

distribution model of the wavelet coefficients of the original image using a generalized Gaussian model and developed the Bayes-Shrink denoising algorithm. Crouse et al. [8] carried out denoising using a hidden Markov tree model, whereas Portilla et al. [9] used a Gaussian scale mixtures model. Şendur and Selesnick [10] proposed a Bivariate model and a corresponding Bishrink algorithm that achieved good image denoising performance. Achim et al. [11] proposed an α -distribution that approximates the distribution of the wavelet coefficients of SAR images. Xie et al. [12] used a finite Gaussian mixture model to approximate the distribution of wavelet coefficients. Although each of these models reflects only a portion of the feature information of the wavelet coefficients, they have contributed to improvements in the denoising of images.

Sparse representation based on an overcomplete dictionary is a novel image representation theory. Various structural features of an image can be effectively captured by using the redundancy of the overcomplete dictionary, ultimately resulting in the image being effectively represented. The K-SVD algorithm [13], which adaptively updates the dictionary based on K -means clustering, has exhibited good performance in image compression and restoration. The SA-DCT approach [14] exploits a shape-adaptive transform on neighbourhoods whose shapes are adaptive to salient image details and thus predominantly contain homogeneous signals. The shape-adaptive transform can achieve a very sparse representation of the true signal in these adaptive neighbourhoods. Dabov et al. developed a block-matching 3D transformation (BM3D) iterative image reconstruction algorithm [15] that is considered one of the best image denoising algorithms in the literature.

This paper combines current multiscale, multiresolution analysis to propose a novel nonsubsamped contourlet transform (NSCT) denoising scheme based on the normal inverse Gaussian (NIG) probability density function (PDF). This proposed scheme uses the NIG distribution to gain the Bayesian maximum *a posteriori* estimator of NSCT coefficients, which are heavy-tailed in nature. First, the parameters of the NIG model are adaptively estimated via a local window that models the intrascale dependency between coefficients. Then, optimal linear interpolation thresholding algorithm named OLI-shrink [16] is utilised. Because the NSCT is translation invariant and has sufficient redundant information, the proposed algorithm can effectively extract the image orientation information and better meet human visual requirements. Experimental results show that the proposed algorithm has better performance than other image denoising methods, including nonsubsamped contourlet transform domain denoising and BLS-GSM [9]. Even though the proposed scheme is a probability-based algorithm, it nevertheless achieves outstanding denoising performance in terms of both peak signal-to-noise ratio (PSNR) and subjective visual quality. We explore the use of the proposed scheme for image denoising and show that it achieves better results than the K-SVD algorithm [13] for all noise levels and performs better than the BM3D algorithm [15] for most noise levels.

The remainder of this paper is organized as follows. Section 2 outlines the basic idea underlying NSCT and the basic image processing scheme. Section 3 discusses the edge coefficient of NSCT statistics modelling and imparts basic knowledge necessary to develop the proposed denoising algorithms, including NIG prior, maximum *a posteriori* probability, parameter estimation, and OLI-Shrink. Experimental results are presented and discussed in Section 4. Finally, concluding remarks are given in Section 5.

2. Nonsubsampled Contourlet Transform (NSCT)

2.1. Overview of NSCT. Wavelet transform is used extensively in image and voice signal processing fields because of its frequency localisation and multiscale and multiresolution properties. However, it only represents point-singularities efficiently in a one-dimensional domain and is less efficient for line-singularities and curve-singularities (edges) in a two-dimensional domain. Wavelet transform also suffers from several fundamental shortcomings, including poor directional selectivity for the anisotropy of its basis function and lack of invariance in shift and orientation [17]. Thus, wavelet transform does not possess optimal properties for two-dimensional signals such as natural images. These two drawbacks of wavelet transform limit its ability to fully capture the directional information in natural images. Consequently, many researchers have begun to focus on methods of expressing image geometry structures more effectively in order to overcome the inability of wavelet transformation to adequately represent the geometric structure information of images. These activities have led to the introduction of ridgelet transform [18], curvelet transform [19], and contourlet transform [20] for singular analysis of two-dimensional or higher images, techniques which can achieve good sparsity for spatially localised details, such as edges and singularities. Because such details are typically abundant in natural images and convey a significant portion of the information embedded therein, these transforms have become useful in image processing [21, 22].

NSCT, proposed by da Cunha et al. in 2006, outperforms the abovementioned transforms [23] in image processing. Its main properties are multiresolution, localisation, directionality, and anisotropy. The directionality property, in particular, permits NSCT to resolve intrinsic directional features that characterise the analysed image. NSCT is based on a nonsubsamped pyramid structure and nonsubsamped directional filter banks, which enables it to realise and develop a flexible, multiscale, multidirectional, and shift-invariable image decomposition that can be efficiently implemented via the *à trous* algorithm. At the core of the NSCT scheme is the nonseparable, two-channel, nonsubsamped filter bank. The nonsubsamped pyramid decomposition structure is realised via a multistage iterative process. An image is decomposed into a two-dimensional, low-frequency subband and a two-dimensional, high-frequency subband using filters. The multistage structure can be realised by performing iterative filtering on the low-frequency subband. Compared to the

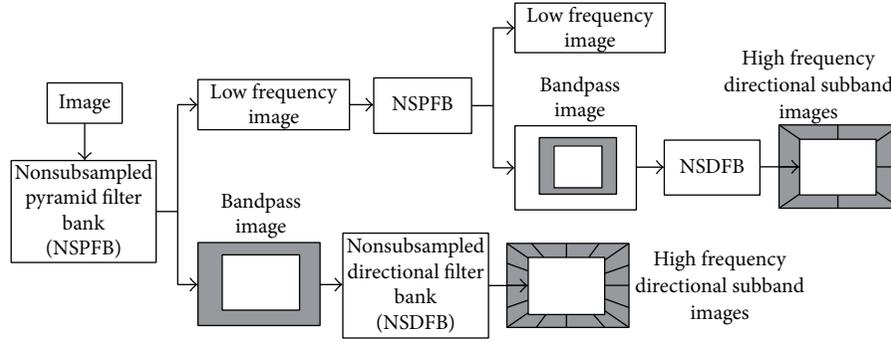


FIGURE 1: NSCT flow graph.

contourlet transformation, the filters of NSCT have better frequency selectivity and regularity, which can better perform subband direction decomposition. In contrast to wavelet decomposition, NSCT is a multiscale, multiresolution, and multidirectional analysis method. Because of the flexibility of the direction decomposition achievable using NSCT, more detailed information about the image can be obtained during the image decomposition process. The resulting transform not only has the multiscale and time-frequency-localisation properties of wavelets but also offers a high degree of directionality and anisotropy. Figure 1 illustrates the NSCT processing scheme [24].

2.2. Image Denoising Method Based on NSCT. The objective of image denoising is to recover an image from its noisy version. In general, a denoising problem can be described as follows. Let $X_{i,j}$ be the original image of size $I \times J$ and let $Y_{i,j}$ be the observation image which has been corrupted by additive white Gaussian noise $\varepsilon_{i,j}$ with zero mean and variance σ^2 . Let (i, j) be the pixel position in the image. Then, assume that the corrupted image satisfies

$$Y_{i,j} = X_{i,j} + \varepsilon_{i,j}, \quad (1)$$

$$i = 1, 2, \dots, I, \quad j = 1, 2, \dots, J, \quad \varepsilon_{i,j} \sim N(0, \sigma^2).$$

The goal of denoising is to construct the optimal approximation of $X_{i,j}$ using the observation data $Y_{i,j}$ to minimize the mean square error (MSE) between the optimal approximation $\hat{X}_{i,j}$ and the original signal $X_{i,j}$:

$$\text{MSE} = \frac{1}{N} E \|\hat{X} - X\|^2 = \frac{1}{N} \sum_{i=1}^I \sum_{j=1}^J E (\hat{X}_{i,j} - X_{i,j})^2, \quad (2)$$

$$N = I \times J.$$

The basic steps in the threshold denoising algorithm are as follows:

- (1) Determine the decomposition level, K , of NSCT and perform NSCT on the noisy image, Y , to obtain the high and low-frequency coefficients of the decomposition.

- (2) Determine the correct threshold value and perform threshold processing on the high-frequency coefficients in the NSCT domain to obtain the new transform coefficients, while leaving the low-frequency coefficients unchanged. The threshold processing includes soft and hard threshold processing.
- (3) Perform inverse NSCT on the processed high-frequency coefficients and low-frequency coefficients to obtain the denoised image estimation, \hat{X} , which is the estimation of the original image, X .

The key issue in threshold denoising is selection of the threshold value, which directly determines the denoising effect. A relatively small threshold value may retain the decomposition coefficients as much as possible and thus retain more detailed information about the image. However, a small threshold value may retain an undesired amount of noise in the denoised result. Conversely, a relatively large threshold value may destroy the high-frequency information of the image and produce a false Gibbs phenomenon in the denoised image.

3. Marginal Statistical Modelling on the NSCT Subband Coefficients

3.1. Normal Inverse Gaussian Distribution. The important features of the image statistics in the NSCT domain is that they are non-Gaussian, have a high kurtosis, sharp central peak, and heavy tails. These features are expected because images often primarily comprise homogeneous regions with some important details such as edges; the homogeneous regions provide coefficients that are close to zero, and the edges provide a small number of coefficients with large magnitudes. The NIG model developed by Barndorff-Nielsen [25] is a normal variance-mean mixture distribution, in which the inverse Gaussian PDF is used as the mixing distribution. In theory, the hybrid model can overcome the disadvantages of the traditional model, which cannot meet the needs of modelling. Because of the flexibility with which parameters can be selected, the hybrid model can describe curves of any shape. Consequently, we chose the NIG model

to describe the distribution of the NSCT coefficients of an image. The probability density function can be defined as

$$f_x(x) = \frac{\alpha\delta}{\pi q(x)} \cdot \exp[p(x)] \cdot K_1[\alpha q(x)], \quad (3)$$

where $p(x) = \delta\sqrt{\alpha^2 - \beta^2} + \beta(x - \mu)$, $q(x) = \sqrt{\delta^2 + (x^2 - \mu^2)}$, and $K_1(\cdot)$ denotes a modified Bessel function of the second kind with index one. As can be seen, the distribution of NIG is specified by the four parameters $(\alpha, \beta, \mu, \delta)$. The flexibility of the values of the four parameters makes the NIG density a suitable model for a variety of unimodal, positive, kurtotic data. Parameter α is the feature factor which controls the steepness of the density; a smaller value for α indicates a slower decay rate with heavier tails. Parameter β controls skewness: for $\beta < 0$, the density is skewed to the left; for $\beta > 0$, the density is skewed to the right; and $\beta = 0$ implies a symmetric density around μ , which is a translation parameter. The δ parameter is scale-like.

For most images, the corresponding decomposition coefficients are generally symmetrical distributions; thus, $\beta = \mu = 0$ is assumed to correspond to NIG. Consequently, the probability density function corresponding to NIG can be simplified as follows:

$$f_x(x) = \frac{\alpha\delta \exp(\alpha\delta)}{\pi} \frac{K_1(\alpha\sqrt{x^2 + \delta^2})}{\sqrt{x^2 + \delta^2}}. \quad (4)$$

3.2. Marginal Distribution of the NSCT Coefficients. In recent years, the Bayes denoising algorithm, which is based on a coefficient's statistical model in the transform domain, has become a hot topic in the field of image denoising. By establishing a reasonable prior probability distribution model for the edge of the coefficients, it estimates the original image using Bayesian theory and balances suppression of noise and retention of signals better than other denoising algorithms. Bayes' algorithm is more flexible than conventional threshold algorithms. Its key to success is to accurately model the prior marginal distribution of coefficients firstly. The main area of research in this case is the marginal distribution statistical models of image NSCT coefficients.

For flexible selection of model parameters, the NIG distribution can accurately model the data with different tails [26]. This paper uses the NIG distribution to fit the marginal statistical distribution of the NSCT coefficients. Because an image typically comprises a smooth area and details such as edges, decomposition coefficients corresponding to smooth regions are, for the most part, close to zero. Only some edges correspond to the high-magnitude transform coefficient. Thus, in the NSCT domain, the image decomposition coefficients show highly non-Gaussian features such as high kurtosis, sharp central peaks, and heavy tails.

We used an image of *Barbara*, which includes texture and structure, to test how well the NIG distribution fits the NSCT coefficients. Figure 2 shows the results obtained. The histogram distribution (solid line) of the third layer subband of NSCT coefficients in the first direction is shown in the figure, with the NIG PDF (dashed line) and Gaussian PDF

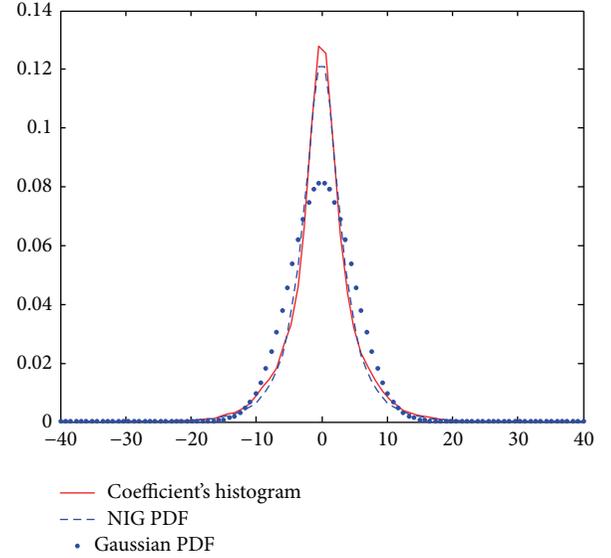


FIGURE 2: Probability density of the NSCT coefficients in a specific subband of a 512×512 pixel *Barbara* image and the NIG PDF and Gaussian PDF fitted to this density.

(dotted line) fitted to this density. As shown in the figure, the NIG PDF can accurately model the coefficient distribution, especially the heavy tail. The distribution demonstrates a sharp peak near to zero and long tails to both sides of the peak, which more closely fit the coefficient's histogram. It also fits the degree of the distribution of the coefficients on the other subbands well. Similar statistical characteristics of the histograms of other test images still hold.

3.3. Bayesian Estimation. Assume that an original image X has been corrupted by additive white Gaussian noise:

$$Y = X + N, \quad (5)$$

where Y denotes the observed noisy image and N denotes the white Gaussian noise. Given Y , denoising attempts to return X as accurately as possible. After NSCT transformation of the observed image, the transformed decomposition coefficients are

$$y = x + n, \quad (6)$$

where y , x , and n denote NSCT coefficients of the noisy image, noise-free original image, and noise, respectively. The purpose of Bayesian denoising is to obtain $\hat{x}(y)$, which is the estimation of x . Bayesian maximum *a posteriori* estimation can be denoted as follows:

$$\hat{x}(y) = \arg \max_x \{f_{x|y}(x | y)\}, \quad (7)$$

where $f_{x|y}(x | y)$ is the conditional density of the observation y given x .

$\hat{x}(y)$ can be calculated using the Bayesian rule:

$$\begin{aligned}\hat{x}(y) &= \arg \max_x \{f_{x|y}(x|y)\} \\ &= \arg \max_x \{f_{y|x}(y|x) \cdot f_x(x)\} \\ &= \arg \max_x \{f_n(y-x) \cdot f_x(x)\}.\end{aligned}\quad (8)$$

These equations allow us to write this estimation in terms of $f_n(\cdot)$ and $f_x(\cdot)$, which are the probability distribution of noise coefficients and the prior distribution of the noise-free image coefficients, respectively. From the assumption on the noise, $f_n(\cdot)$ is a zero-mean Gaussian function with variance σ_n^2 ; that is,

$$f_n(n) = \frac{1}{\sqrt{2\pi}\sigma_n} \cdot \exp\left(-\frac{n^2}{2\sigma_n^2}\right).\quad (9)$$

Solving (8) by using (9) and performing a logarithm on the independent variable, (8) can be written as

$$\hat{x}(y) = \arg \max_x \left[-\frac{(y-x)^2}{2\sigma_n^2} + \zeta(x) \right],\quad (10)$$

where $\zeta(x) = \ln f_x(x)$ is a convex, differentiable function. Let the first derivative of $-(y-x)^2/2\sigma_n^2 + \zeta(x)$ be zero. Then, $\hat{x}(y)$ can be computed to obtain the maximum *a posteriori*; thus

$$\frac{y-\hat{x}}{\sigma_n^2} + \zeta'(\hat{x}) = 0.\quad (11)$$

Because $\zeta'(x)$ is discontinuous and singular in the vicinity of zero and the symbol of $\hat{x}(y)$ is different from y , (11) cannot be solved in closed form. Bhuiyan et al. [27] developed an estimated solution that overcomes this drawback. It is defined as

$$\hat{x}(y) = \text{sign}(y) \cdot \max(|y| - \sigma_n^2 B, 0),\quad (12)$$

where

$$B = \left| \frac{2y}{\delta^2 + y^2} + \frac{\alpha y}{\sqrt{\delta^2 + y^2}} \cdot \frac{K_0(\alpha\sqrt{\delta^2 + y^2})}{K_1(\alpha\sqrt{\delta^2 + y^2})} \right|.\quad (13)$$

Parameters α and δ are estimated from every subband. As a result, (12) is adaptive to every subband and can remove noise from all the subbands adaptively. Equation (12) is also similar to the classical soft threshold function. The threshold value is $\sigma_n^2 B$ and can be changed depending on the values of the coefficients. The noise-based coefficient is greater with larger threshold values. Conversely, the signal-based coefficient is larger with smaller threshold values.

3.4. Parameter Estimation. Image denoising via (12) needs to estimate the NIG distribution parameters α and δ and the noise variance σ_n^2 . The parameters used in the experiments

can be estimated according to the actual distribution of different subband coefficients. For different noise levels, these parameters can be used to obtain different degrees of threshold processing given different decomposition coefficients.

Let \hat{k}_1 , \hat{k}_2 , \hat{k}_3 , and \hat{k}_4 denote the one to four-order cumulants of the noisy coefficients, respectively. Thence, the skewness and kurtosis of the coefficients of the clear image are $\gamma_3 = \hat{k}_3/(\hat{k}_2)^{3/2}$ and $\gamma_4 = \hat{k}_4/(\hat{k}_2)^2$, respectively. At this time, the parameters α and δ can be estimated using the following two equations:

$$\begin{aligned}\delta &= \sqrt{\xi \cdot \hat{k}_2 (1 - \rho^2)}, \\ \alpha &= \frac{\xi}{\left(\delta \sqrt{1 - \rho^2}\right)},\end{aligned}\quad (14)$$

where $\xi = 3 \cdot (\gamma_4 - 4\gamma_3^2/3)^{-1}$ and $\rho = \gamma_3 \sqrt{\xi}/3$.

3.5. Thresholding Algorithm. Denoising by wavelet is performed by the thresholding algorithm, in which coefficients smaller than a specific value, or threshold, are cancelled. It can also be utilised in other transform domains, such as the NSCT domain. Hard and soft thresholding are two popular methods. In hard thresholding algorithms, any coefficient, y , less than or equal to the threshold, λ , is replaced with zero. That is,

$$\delta_\lambda^H(y) = \begin{cases} 0, & |y| \leq \lambda \\ y, & |y| > \lambda. \end{cases}\quad (15)$$

In soft thresholding algorithms, however, any coefficient, y , less than the threshold λ is replaced with zero, and the others are modified by subtracting the threshold value λ in accordance with the following equation:

$$\delta_\lambda^S(y) = \begin{cases} 0, & |y| \leq \lambda \\ \text{sign}(y)(|y| - \lambda), & |y| > \lambda. \end{cases}\quad (16)$$

Soft thresholding is more efficient and yields more visually pleasing images than hard thresholding, but soft thresholding does not use the optimal value for modification of large coefficients. In order to overcome this limitation, Fathi and Naghsh-Nilchi [16] developed a new thresholding algorithm (OLI-Shrink) which uses optimal linear interpolation between each coefficient and its corresponding subband mean to modify dominant coefficients:

$$\delta_\lambda^{\text{OLI}}(y) = \begin{cases} 0, & |y| \leq \lambda \\ y - \eta(y - \mu), & |y| > \lambda, \end{cases}\quad (17)$$

where μ is the mean value of the coefficient of the corresponding subband, $\lambda = \sigma^2 B$ as in (12), and η is computed as follows:

$$\eta = \frac{\sigma_n^2}{\sigma_x^2 + \sigma_n^2} \cong \frac{\sigma_n^2}{\sigma_y^2}.\quad (18)$$

Here, the true value x is estimated by a weighted linear interpolation of the unconditional mean of x and the observed value y . Even if this formula is derived on the assumption that the transform domain coefficients in a subband of a natural image have Gaussian PDFs, it can still function efficiently.

3.6. Monte-Carlo Estimates of NSCT Domain Noise Coefficient Variance. The nonorthogonal nature of NSCT may lead to different noise variances in different directional subbands. Thus, the Monte-Carlo method is used to obtain the noise coefficient variance $\sigma_n^2(k)$ of NSCT, which corresponds to each subband at each scale. In some image denoising applications, the value of the input noise variance is known or can be measured based on information other than the corrupted image. In other cases, we can estimate the noise variance by applying the robust median estimator on the HHI subband's coefficients ($Y_{i,j}^{\text{HHI}}$), as outlined by Donoho [3]:

$$\hat{\sigma}_n = \frac{\text{median}(|Y_{i,j}^{\text{HHI}}|)}{0.6745}. \quad (19)$$

The specific steps are as follows.

Step 1. Perform orthogonal wavelet transform on the noisy image and estimate the noise standard deviation, $\hat{\sigma}_n$, using (19).

Step 2. Construct a white Gaussian noise image the same size as the original image with mean zero and variance $\hat{\sigma}_n^2$.

Step 3. Perform NSCT transformation on the noise image and obtain the noise NSCT coefficients variance $\hat{\sigma}_n^2(k)$ in each high-frequency subband.

Step 4. Go to Step 2 and repeat the above steps five times. The final coefficient variance, $\sigma_n^2(k)$, is obtained by averaging the coefficient variances obtained.

3.7. Detailed Implementation. In summary, we apply the NIG model as a prior model of NSCT coefficients to estimate denoised NSCT coefficients and employ OLI-Shrink to guarantee a more gently thresholded effect. The proposed method can be summarized as follows.

Step 1. Perform NSCT decomposition to obtain a series of NSCT coefficients, y .

Step 2. Estimate the noise variance in each high-frequency subband using the method outlined in Section 3.6.

Step 3. For each subband in each level higher than two, compute the threshold value and statistical parameters:

- (a) the terms δ and α using (14);
- (b) the threshold value $\lambda = \sigma_n^2 B$ using (13), where the σ_n^2 used here is equal to $\sigma_n^2(k)$ in corresponding subbands;
- (c) the subband's variance (σ_y^2);

(d) the subband's mean (μ);

(e) the term η using (18).

Step 4. Threshold the coefficients of all subbands using (17);

Step 5. Perform inverse NSCT to reconstruct the denoised image.

4. Experimental Results

To investigate the feasibility and the effectiveness of the proposed image denoising method in removing additive Gaussian white noise, we first conducted experiments on the grayscale images *Lena* and *Barbara*, each measuring 512×512 pixels. The images were obtained from <http://www.cs.tut.fi/~foi/GCF-BM3D>, and we verified that they were the same as the images used by Dabov et al. [15]. Using these images, we redid Dabov et al.'s entire denoising experiment. The original images were corrupted with zero mean and Gaussian white noise, with the standard deviation, σ , of the noise varying over the range 5 to 100. For comparison, we also applied several existing methods to denoise the same images. BivShrink [10] was implemented using software obtained from <http://eeweb.poly.edu/iselesni/WaveletSoftware/denoise2.html>, which describes the implementation of this algorithm using both the separable DWT and the complex dual-tree DWT. We chose the complex dual-tree DWT method because it gives better denoised results. The BLS-GSM denoising method [9] was implemented using software obtained from <http://decsai.ugr.es/~javier/denoise/index.html>. The SA-DCT denoising method [14] was implemented using software obtained from <http://www.cs.tut.fi/~foi/SA-DCT>. The K-SVD [13] denoising method was implemented using software obtained from <http://www.cs.technion.ac.il/~elad/software/>, and the image reconstruction algorithm [15] based on block matching and 3D filtering was implemented using software obtained from <http://www.cs.tut.fi/~foi/GCF-BM3D>. Ram et al. [28] recently proposed an image processing scheme based on reordering of patches. They applied their proposed approach to image denoising and showed that it generally performs better than the BM3D algorithm for high noise levels. Thus, we implemented this denoising method using software obtained from <http://www.cs.technion.ac.il/~elad/software/>.

The decomposition level of the proposed method was set at five and the decomposed direction from coarse scale to fine scale ranged over the values 4, 8, 8, 16, and 16. The local window size was 11×11 when estimating local parameters. The quantitative analysis results obtained are listed in Table 1. Each denoising method outlined above was quantitatively measured in terms of peak signal-to-noise ratio (PSNR). The details of the denoised images are given in Figures 3 and 5 to help the reader observe the qualitative differences between the different methods.

As shown in Table 1, the proposed method achieved the best results among all the denoising methods, except for some of the results from BM3D and Ram's method, with an average increase in PSNR of approximately 0.29 dB compared to Ram's method and 0.17 dB compared to BM3D. In particular, when compared with denoised results obtained by K-SVD

TABLE 1: Denoising results (PSNR in dB) of the various denoising methods with different noise standard deviations, σ . The best result for each image and setting is displayed in boldface.

| Image | Method | σ /PSNR | | | | | | | |
|----------------|-----------|----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | | 5/34.16 | 10/28.14 | 15/24.61 | 20/22.11 | 25/20.18 | 50/14.16 | 75/10.63 | 100/8.14 |
| <i>Lena</i> | BivShrink | 38.05 | 35.32 | 33.59 | 32.33 | 31.36 | 28.24 | 26.44 | 25.21 |
| | BLS-GSM | 38.56 | 35.68 | 33.96 | 32.72 | 31.75 | 28.69 | 26.96 | 25.83 |
| | SA-DCT | 38.54 | 35.58 | 33.87 | 32.63 | 31.66 | 28.60 | 26.76 | 25.51 |
| | K-SVD | 38.61 | 35.51 | 33.71 | 32.40 | 31.37 | 27.85 | 25.82 | 24.55 |
| | Ram's | 38.33 | 35.39 | 33.81 | 32.68 | 31.78 | 29.01 | 27.24 | 26.00 |
| | BM3D | 38.72 | 35.93 | 34.27 | 33.05 | 32.08 | 28.86 | 27.02 | 25.57 |
| | Proposed | 39.13 | 36.18 | 34.52 | 33.32 | 32.37 | 29.37 | 27.57 | 26.26 |
| <i>Barbara</i> | BivShrink | 36.74 | 33.38 | 31.27 | 29.75 | 28.61 | 25.23 | 23.53 | 22.48 |
| | BLS-GSM | 37.85 | 34.10 | 31.93 | 30.38 | 29.19 | 25.50 | 23.67 | 22.65 |
| | SA-DCT | 37.49 | 33.50 | 31.39 | 30.00 | 28.95 | 25.43 | 23.52 | 22.49 |
| | K-SVD | 38.10 | 34.45 | 32.42 | 30.85 | 29.57 | 25.48 | 22.97 | 21.87 |
| | Ram's | 37.75 | 34.41 | 32.70 | 31.50 | 30.54 | 27.39 | 25.54 | 24.24 |
| | BM3D | 38.31 | 34.98 | 33.11 | 31.78 | 30.72 | 27.17 | 25.10 | 23.49 |
| | Proposed | 38.36 | 34.82 | 32.84 | 31.45 | 30.38 | 27.10 | 25.26 | 24.01 |

Bold number indicates best performing method in separate column.

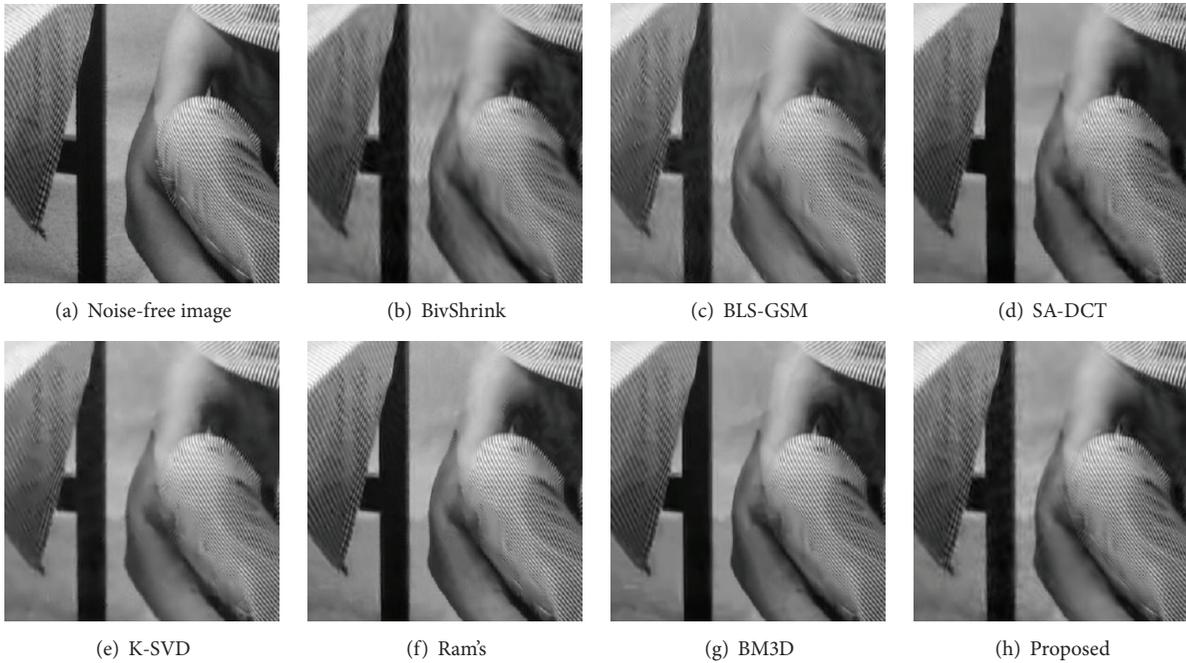


FIGURE 3: Fragments of noisy ($\sigma = 25$, PSNR = 20.18 dB) *Barbara* image and corresponding estimates with different methods.

method, the average PSNR increase for the proposed method was 1.09 dB. For image *Barbara*, Ram's method actually performed better than the BM3D algorithm for high noise levels. Ram's method also had a higher PSNR value than BM3D for the *Lena* image at high noise levels, but it was inferior to our proposed method for all noise levels.

The denoising performance of our proposed algorithm is illustrated in Figure 3, which depicts fragments of a few noisy ($\sigma = 25$) modifications of the original *Barbara* image and fragments of various image denoising results.

In Figure 3, it can be seen that our proposed method exhibits outstanding performance which is superior to the other methods, especially in terms of retaining the image details, such as the small stripes in the tablecloth. BM3D and the proposed method produce fewer artefacts and better preserve the edges and other details than the other methods. The homogeneous region is smoother and the texture line on the tablecloth and trousers is clearer than the results from the other methods. The denoised images in Figures 3(g) and 3(h) are relatively mild, have fewer scratch effects,

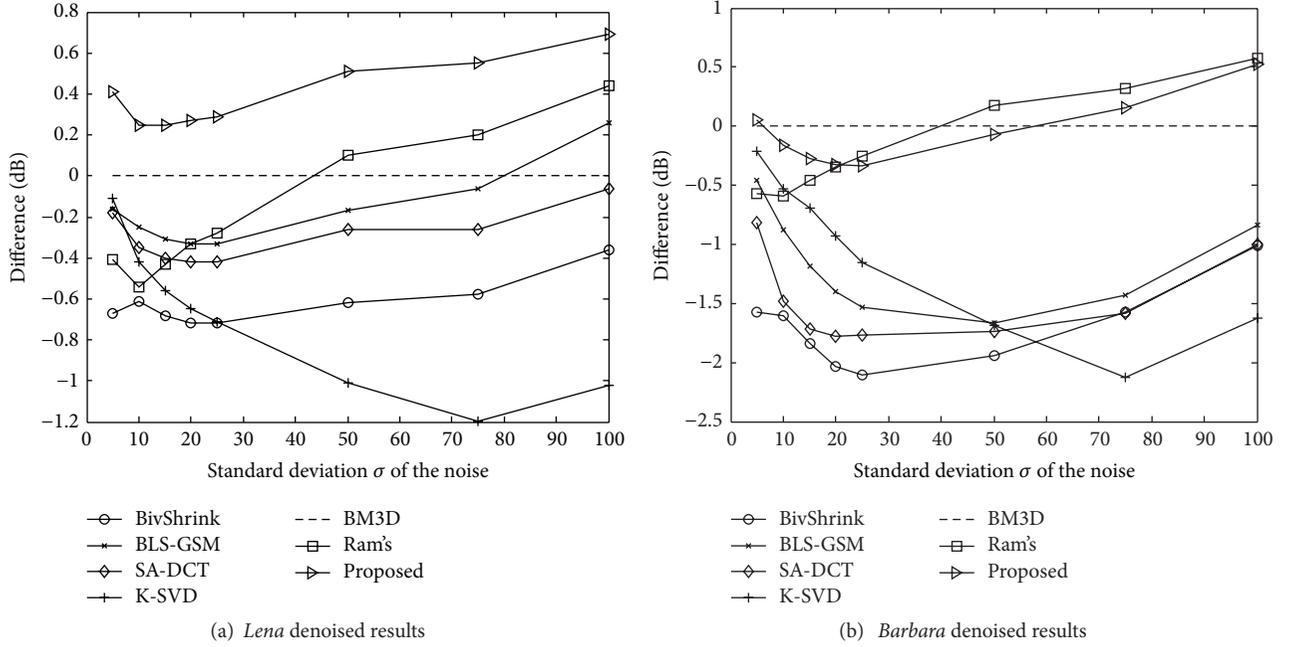


FIGURE 4: Performance gain in terms of PSNR values attained by various schemes versus the proposed scheme.

show most of the texture features clearly, and retain their original geometric features, such as the clear checkerboard tablecloth pattern. The resulting image following denoising by BLS-GSM (Figure 3(c)) appears to have better overall effects; however, close inspection shows that the image has been smoothed too much and, as a consequence, has lost much detailed information, information which is very important in image denoising. Figures 3(d) and 3(e), obtained from SA-DCT and K-SVD, have excessive smoothing and insufficient contrast, and some small details are missing. Ram's method appears to be efficient in homogeneous regions but is less effective in the detailed regions. The evaluation values in Table 1 show that the methods applied to produce Figures 3(b)–3(f) are significantly worse than BM3D and the proposed method. In conclusion, it is clear that our denoised method exhibits outstanding performance and preserves detailed information, especially texture information.

In order to better understand the numerical results, we further compared the gain of BivShrink, BLS-GSM, SA-DCT, K-SVD, Ram's method, and the proposed method, normalised with respect to BM3D, as shown in Figure 4. In the figure, the superiority of the proposed method is clear. In the case of image *Lena*, the proposed method outperforms the other methods and provides significant improvements over BM3D. Compared with BM3D, the performance of Ram's method is slightly inferior when the standard deviation is small. BLS-GSM and SA-DCT have similar results, which are better than that of BivShrink. The performance of the proposed method is as expected. Image *Barbara* comprises similar image textures and can therefore provide more information to BM3D and Ram's method to train their datasets. As a result, their output is a bit better than that of the proposed method at the start (Figure 4(b)).

BM3D is based on block structure [29], which tends to oversmooth edge pixels after denoising. As a result, its reconstructed image shows obvious blocking artefacts under certain conditions, especially when the noise variance is relatively large. Figures 5(a) and 5(b) show fragments of the reconstructed *Barbara* images obtained using BM3D and the proposed scheme, respectively. When the noise standard deviation is 50 for the *Barbara* image, the denoised results with BM3D and the proposed scheme have virtually the same PSNR value. However, it is clear that the stripe information in Figure 5(a) has essentially been replaced by smooth areas, whereas the stripe is still clearly visible in Figure 5(b).

From the above results, it is clear that the more competitive denoising methods are Ram's method, BM3D, and the proposed method. To provide a more thorough comparison of these three methods, we assessed their performance on a test set containing noisy versions of seven images with eight different noise levels. For the proposed method, four-level decomposition and the decomposed direction from coarse scale to fine scale were 4, 8, 16, and 16 for the images *House* and *Peppers*, with sizes 256×256 pixels. The PSNR values for the results obtained using the three schemes are listed in Table 2.

The results obtained by our scheme are inferior to those of Ram's method for the images *House*, *Peppers*, and *Montage* in most cases. The PSNR values for the images *Fingerprint*, *Man*, and *Couple* are significantly better than those obtained by BM3D and Ram's method. Our method performed as expected for *Boats* and *Hill*, providing the highest PSNR values. Further, our results are better than those achieved by Ram's algorithm by approximately 0.31 dB on average for the entire number lists belonging to Ram's algorithm and our scheme, respectively, in Table 2. Further, our scheme

TABLE 2: Denoising results (PSNR in dB) for noisy versions of seven images, obtained with BM3D, Ram's method, and the proposed scheme. The best result for each image and noise level is displayed in boldface.

| Image | Method | σ /PSNR | | | | | | | |
|--------------------|----------|----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | | 5/34.16 | 10/28.14 | 15/24.61 | 20/22.11 | 25/20.18 | 50/14.16 | 75/10.63 | 100/8.14 |
| <i>Boats</i> | Ram's | 37.07 | 33.68 | 31.91 | 30.69 | 29.72 | 26.72 | 25.03 | 23.92 |
| | BM3D | 37.28 | 33.92 | 32.14 | 30.88 | 29.91 | 26.64 | 24.96 | 23.74 |
| | Proposed | 37.50 | 34.07 | 32.25 | 31.00 | 30.05 | 27.23 | 25.67 | 24.57 |
| <i>Fingerprint</i> | Ram's | 36.17 | 32.00 | 29.83 | 28.37 | 27.33 | 24.10 | 22.44 | 21.46 |
| | BM3D | 36.51 | 32.46 | 30.28 | 28.81 | 27.70 | 24.36 | 22.68 | 21.33 |
| | Proposed | 36.36 | 32.62 | 30.58 | 29.16 | 28.10 | 24.91 | 23.12 | 21.87 |
| <i>Man</i> | Ram's | 37.38 | 33.67 | 31.77 | 30.48 | 29.55 | 26.74 | 25.17 | 24.15 |
| | BM3D | 37.82 | 33.98 | 31.93 | 30.59 | 29.62 | 26.59 | 25.10 | 23.97 |
| | Proposed | 37.78 | 34.17 | 32.29 | 31.04 | 30.11 | 27.42 | 25.90 | 24.82 |
| <i>Couple</i> | Ram's | 37.12 | 33.60 | 31.74 | 30.46 | 29.42 | 26.39 | 24.69 | 23.50 |
| | BM3D | 37.52 | 34.04 | 32.11 | 30.76 | 29.72 | 26.38 | 24.63 | 23.37 |
| | Proposed | 37.36 | 33.80 | 31.90 | 30.62 | 29.66 | 26.83 | 25.27 | 24.19 |
| <i>Hill</i> | Ram's | 36.92 | 33.48 | 31.77 | 30.64 | 29.76 | 27.17 | 25.61 | 24.58 |
| | BM3D | 37.14 | 33.62 | 31.86 | 30.72 | 29.85 | 27.08 | 25.58 | 24.45 |
| | Proposed | 37.31 | 33.93 | 32.21 | 31.07 | 30.22 | 27.72 | 26.31 | 25.29 |
| <i>House</i> | Ram's | 38.76 | 35.82 | 34.38 | 33.33 | 32.55 | 29.77 | 27.68 | 26.20 |
| | BM3D | 39.83 | 36.71 | 34.94 | 33.77 | 32.86 | 29.37 | 27.20 | 25.50 |
| | Proposed | 39.44 | 36.13 | 34.27 | 32.98 | 32.00 | 28.88 | 27.00 | 25.64 |
| <i>Peppers</i> | Ram's | 37.62 | 34.22 | 32.33 | 31.02 | 29.92 | 26.72 | 24.75 | 23.32 |
| | BM3D | 38.12 | 34.68 | 32.70 | 31.29 | 30.16 | 26.41 | 24.48 | 22.91 |
| | Proposed | 37.64 | 34.12 | 32.18 | 30.83 | 29.80 | 26.65 | 24.81 | 23.51 |

Bold number indicates best performing method in separate column.

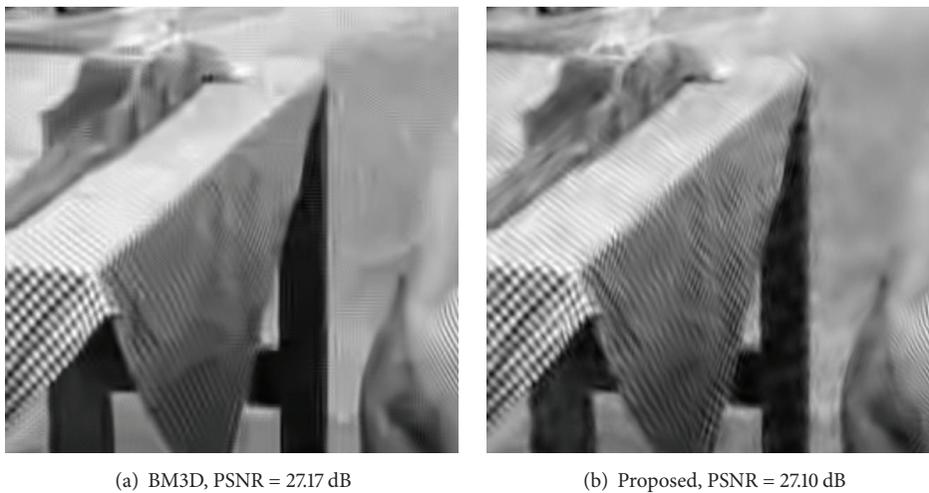


FIGURE 5: Fragments of the grayscale *Barbara* image denoised by BM3D and our proposed algorithm for noise with $\sigma = 50$.

outperforms BM3D by more than 0.85 dB (Man, $\sigma = 100$) and 0.16 dB on average for all results obtained by the BM3D method as listed in Table 2. In order to facilitate a more comprehensive comparison, the noisy and denoised images obtained with our proposed scheme for $\sigma = 25$ are shown in Figures 6 and 7.

An image has a specific structure and its pixels have strong affiliations. These affiliations reflect the configuration

information in the visual scene. In order to better reflect the structural similarity between two images, Wang et al. [30] developed an image quality assessment method based on structural distortion called the structural similarity (SSIM) index. There are also several similar structural information based image quality metrics (IQMs), including low-level feature similarity induced full reference image quality assessment metric, namely FSIM [31] and sparse feature



FIGURE 6: Denoising results (PSNR) for the images *Lena*, *Barbara*, *Boat*, and *House* ($\sigma = 25$, PSNR = 20.18 dB): (a) noisy images; (b) Ram's results; (c) BM3D results; and (d) our proposed scheme's results (PSNR scores are given below each image).

fidelity (SFF) [32]. The closer the IQMs are to one, the higher the structural similarity between the two images is. In image denoising, the IQMs indicate the ability of the denoising algorithm to maintain the image structure. Table 3 shows the SSIM, FSIM, and SFF index values for image

Barbara following denoised processing by BM3D and the proposed method. The average PSNR value for the proposed method is smaller than that of BM3D for image *Barbara* by approximately 0.06 dB in Table 1. However, as the SSIM index listed in Table 3 shows, the denoising results for the proposed



FIGURE 7: Fragments of noisy ($\sigma = 25$, PSNR = 20.18 dB) grayscale images and their corresponding estimates: (a) noisy images; (b) Ram's results; (c) BM3D results; and (d) our proposed scheme's results (PSNR scores are given below each image).

TABLE 3: Image quality metrics for image *Barbara* following denoising by BM3D and the proposed method.

| | σ | 2 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 50 | 75 | 100 | Average |
|------|----------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| SSIM | BM3D | 0.9970 | 0.9891 | 0.9767 | 0.9650 | 0.9530 | 0.9406 | 0.9272 | 0.9123 | 0.8706 | 0.8013 | 0.7364 | 0.9154 |
| | Proposed | 0.9971 | 0.9899 | 0.9771 | 0.9644 | 0.9517 | 0.9391 | 0.9264 | 0.9140 | 0.8773 | 0.8204 | 0.7695 | 0.9206 |
| FSIM | BM3D | 0.9983 | 0.9932 | 0.9840 | 0.9757 | 0.9676 | 0.9600 | 0.9526 | 0.9446 | 0.9244 | 0.8917 | 0.8618 | 0.9504 |
| | Proposed | 0.9983 | 0.9932 | 0.9836 | 0.9741 | 0.9652 | 0.9569 | 0.9491 | 0.9418 | 0.9217 | 0.8933 | 0.8697 | 0.9497 |
| SFF | BM3D | 0.9997 | 0.9987 | 0.9965 | 0.9943 | 0.9921 | 0.9899 | 0.9876 | 0.9851 | 0.9787 | 0.9675 | 0.9553 | 0.9859 |
| | Proposed | 0.9997 | 0.9988 | 0.9966 | 0.9943 | 0.9920 | 0.9896 | 0.9874 | 0.9852 | 0.9786 | 0.9678 | 0.9569 | 0.9861 |

Bold number indicates best performing method in separate column for each image quality metrics.

method are superior to those for BM3D in maintaining the original image structure when the value of σ is below 15 or above 30, and it outperforms the average for σ in the range 2–100. For FSIM and SFF index, familiar results can be given in Table 3. Sometimes they are equal to each other; sometimes they alternately outperform the other side.

5. Conclusion

In this paper, we proposed a new image processing method based on the Bayesian framework that uses the key facets of the Bayesian denoising method to correctly estimate the distribution model of the decomposition coefficients. The proposed method comprises a nonsubsampling contourlet transform denoising algorithm based on the normal inverse Gaussian prior model, which has parameters that can be selected flexibly and changed adaptively. The model can accurately describe the sharp peak of the NSCT decomposition coefficients at zero and heavy-tailed characteristics distributed on both sides symmetrically; consequently, Bayesian denoising can be used to effectively remove noise from images. The results of experiments conducted indicate that the proposed method can effectively eliminate Gaussian white noise. Further, visual and quantitative evaluations show that it performs significantly better than existing state-of-the-art image denoising methods.

The NSCT's high redundancy consequentially leads to an overloading computational cost of proposed method, and this high computational cost may limit proposed method's real time application in practice. Even if quantitative comparison of various denoising methods' computation efficiency is not our study emphasis, how to reduce the complexity of the algorithm and improve the operational efficiency becomes the next focus of the study. In future, we will try to transplant proposed method on other programming platforms such as C++ and adopt parallel compiling technique to improve running efficiency. This proposed method also can be naturally extended in other image processing fields, such as image inpainting, image restoration, and image fusion.

Conflict of Interests

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

Acknowledgments

This research was supported by the National Natural Science Foundation of China (no. 61379010), National Natural Science Foundation of China (no. 61502219), and Postdoctoral Science Foundation of China (no. 2015M582697). The authors would like to thank the editor and the anonymous reviewers for their valuable and insightful comments on this paper.

References

- [1] P. Qiu and P. S. Mukherjee, "Edge structure preserving image denoising," *Signal Processing*, vol. 90, no. 10, pp. 2851–2862, 2010.
- [2] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, Amsterdam, The Netherlands, 3rd edition, 2009.
- [3] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995.
- [4] F. Abramovich, T. Sapatinas, and B. W. Silverman, "Wavelet thresholding via a Bayesian approach," *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, vol. 60, no. 4, pp. 725–749, 1998.
- [5] A. Pižurica, W. Philips, I. Lemahieu, and M. Acheroy, "A joint inter- and intrascale statistical model for Bayesian wavelet based image denoising," *IEEE Transactions on Image Processing*, vol. 11, no. 5, pp. 545–557, 2002.
- [6] S. Tan, L. Jiao, and I. A. Kakadiaris, "Wavelet-based Bayesian image estimation: from marginal and bivariate prior models to multivariate prior models," *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 469–481, 2008.
- [7] S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1532–1546, 2000.
- [8] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 886–902, 1998.
- [9] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1338–1351, 2003.
- [10] L. Şendur and I. W. Selesnick, "Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency," *IEEE Transactions on Signal Processing*, vol. 50, no. 11, pp. 2744–2756, 2002.

- [11] A. Achim, P. Tsakalides, and A. Bezerianos, "SAR image denoising via Bayesian wavelet shrinkage based on heavy-tailed modeling," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 8, pp. 1773–1784, 2003.
- [12] H. Xie, L. E. Pierce, and F. T. Ulaby, "SAR speckle reduction using wavelet denoising and Markov random field modeling," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 10, pp. 2196–2212, 2002.
- [13] 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.
- [14] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1395–1411, 2007.
- [15] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [16] A. Fathi and A. R. Naghsh-Nilchi, "Efficient image denoising method based on a new adaptive wavelet packet thresholding function," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 3981–3990, 2012.
- [17] I. W. Selesnick, R. G. Baraniuk, and N. Kingsbury, "The dual-tree complex wavelet transforms—a coherent framework for multiscale signal and image processing," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 123–151, 2005.
- [18] M. N. Do and M. Vetterli, "The finite ridgelet transform for image representation," *IEEE Transactions on Image Processing*, vol. 12, no. 1, pp. 16–28, 2003.
- [19] J.-L. Starck, E. J. Candès, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 670–684, 2002.
- [20] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091–2106, 2005.
- [21] M. Dong, J. Zhang, and Y. Ma, "Image denoising via bivariate shrinkage function based on a new structure of dual contourlet transform," *Signal Processing*, vol. 109, pp. 25–37, 2015.
- [22] H. Zhong, K. Ma, and Y. Zhou, "Modified BM3D algorithm for image denoising using nonlocal centralization prior," *Signal Processing*, vol. 106, pp. 342–347, 2015.
- [23] 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.
- [24] K. Zhang, Y. H. Sheng, and H. Y. Lv, "Stereo matching cost computation based on nonsubsampling contourlet transform," *Journal of Visual Communication and Image Representation*, vol. 26, pp. 275–283, 2015.
- [25] O. E. Barndorff-Nielsen, "Normal inverse Gaussian distributions and stochastic volatility modelling," *Scandinavian Journal of Statistics*, vol. 24, no. 1, pp. 1–13, 1997.
- [26] X. Zhang and X. L. Jing, "Image denoising in contourlet domain based on a normal inverse Gaussian prior," *Digital Signal Processing*, vol. 20, no. 5, pp. 1439–1446, 2010.
- [27] M. I. H. Bhuiyan, M. O. Ahmad, and M. N. S. Swamy, "Wavelet-based despeckling of medical ultrasound images with the symmetric normal inverse Gaussian prior," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, pp. I-721–I-724, IEEE, Honolulu, Hawaii, USA, April 2007.
- [28] I. Ram, M. Elad, and I. Cohen, "Image processing using smooth ordering of its patches," *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2764–2774, 2013.
- [29] Q. S. Lian and S. Z. Chen, "Sparse image representation using the analytic Contourlet transform and its application on compressed sensing," *Acta Electronica Sinica*, vol. 36, no. 6, pp. 1293–1298, 2010.
- [30] 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.
- [31] 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.
- [32] H.-W. Chang, H. Yang, Y. Gan, and M.-H. Wang, "Sparse feature fidelity for perceptual image quality assessment," *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 4007–4018, 2013.

Research Article

Single Image Superresolution Using Maximizing Self-Similarity Prior

Jianhong Li,¹ Yarong Wu,² Xiaonan Luo,^{1,3} Chengcai Leng,⁴ and Bo Li⁴

¹National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou 510006, China

²Science and Technology Department, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China

³Beijing Key Laboratory of Multimedia and Intelligent Software Technology, College of Metropolitan Transportation, Beijing University of Technology, Beijing 100124, China

⁴School of Mathematics and Information Science, Nanchang Hangkong University, Nanchang 330063, China

Correspondence should be addressed to Xiaonan Luo; lnslxn@mail.sysu.edu.cn

Received 5 September 2015; Revised 10 November 2015; Accepted 12 November 2015

Academic Editor: Daniel Zaldivar

Copyright © 2015 Jianhong Li 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.

Single image superresolution (SISR) requires only one low-resolution (LR) image as its input which thus strongly motivates researchers to improve the technology. The property that small image patches tend to recur themselves across different scales is very important and widely used in image processing and computer vision community. In this paper, we develop a new approach for solving the problem of SISR by generalizing this property. The main idea of our approach takes advantage of a generic prior that assumes that a randomly selected patch in the underlying high-resolution (HR) image should visually resemble as much as possible with some patch extracted from the input low-resolution (LR) image. Observing the proposed prior, our approach deploys a cost function and applies an iterative scheme to estimate the optimal HR image. For solving the cost function, we introduce Gaussian mixture model (GMM) to train on a sampled data set for approximating the joint probability density function (PDF) of input image with different scales. Through extensive comparative experiments, this paper demonstrates that the visual fidelity of our proposed method is often superior to those generated by other state-of-the-art algorithms as determined through both perceptual judgment and quantitative measures.

1. Introduction

The task of single image superresolution (SISR) is concerned with generating a high-resolution (HR) image from an input image of low resolution (LR). Since the same LR image may hypothetically correspond to multiple HR images, the solution space is inherently ambiguous. To resolve the ambiguity, existing methods often rely on certain image priors for selecting the preferred HR reconstruction result.

Existing SISR methods can be broadly classified into three main categories: interpolation-based methods [1–4], reconstruction-based methods [5–9], and learning-based methods [10–16]. Interpolation-based methods usually apply some basic function or interpolation kernel to determine the unknown pixels in the HR version. Overall, these methods are relatively easy to implement. However, they tend to perform inefficiently near image edges and produce blurring artifacts.

Reconstruction-based methods derive HR images via imposing constraints according to prior knowledge of the inverse problem [6, 7]. Shan et al. [6] design a feedback control framework, which uses a continuous function to fit a gradient density distribution for the input image as a prior for the HR reconstruction process. However, these methods often produce undesirable artifacts in the HR results, especially along salient edges.

The basic assumption of the learning-based methods is that the high-frequency details lost in a reconstructed HR image can be learnt from a set of low- and high-resolution image pairs. This category of methods generates an HR image from a single LR image by referring to pairs of low- and high-resolution images as the training data. Extensive research results have demonstrated the promising potential of the approach. Freeman et al. [11, 15] proposed a learning based method applicable for processing generic images where

the mapping from LR to HR versions of an image is captured by a Markov random field inferred through belief propagation. Chang et al. [17] extended their method by applying the manifold learning theory onto the correspondence between the HR image patch space and the LR image patch space. The method generates HR patches as a linear combination of its nearest neighboring patches. However, the fixed number of nearest neighbors could cause over- or underfitting. To conquer this shortcoming, Yang et al. [13, 18, 19] proposed an approach that generates sparse linear combinations using a compact dictionary. The computed sparse representation adaptively selects the relevant patches in the dictionary to represent each patch. It is noted that constructing a compact dictionary suitable for the application is a time-consuming process. These methods severely depend on the quality and relevance of the image training data set. It tends to produce obvious artifacts and generate unwanted noise in its HR generation results.

Some recent studies show that natural images generally possess a great amount of self-similarities [14, 20–22], that is, local image structures tend to recur within and across different image scales [14, 20–22], and image superresolution can be regularized taken into account these self-similar examples instead of some external database [20–22]. Particularly, Yang et al. [14] use self-examples within and across multiple image scales to regularize the ill-posed classical superresolution problem. The method extends the example-based superresolution framework with self-examples and iteratively upscale the image. They show that the local self-similarity assumption for natural images holds better for small upscaling factors and the patch search can be conducted in a restricted local region, allowing very fast practical implementation. However, the main problem rests with the fact that the reconstructed edges are usually too sharp to look natural.

Our work is inspired by the observation that almost all small image patches usually reappear themselves at other positions in natural images or its different scales. In conformity with the above observation, the basic assumption we design in this study is that there should at least exist one resemble patch in the input low resolution image for a patch randomly extracted from the well-reconstructed HR image.

The main contributions of this paper include the following:

- (1) Inspired by the idea that each patch in the reconstructed SR result should resemble patches in the original input image, we propose a new algorithm that infers HR images from their LR versions through multiresolution self-similarity maximization. The results of the new method outperform those generated by the state-of-the-art algorithms through comparative experiments.
- (2) For finding the optimal HR image, we introduce GMM to train on a patch set for approximating the joint probability density function of input image with different scales and uses an iterative scheme to resolve the cost function. This framework is easy to be adapted to other problems such as image deblurring and image denoising.

The rest of this paper is organized as follows. We first state the study motivation in Section 2. Section 3 then introduces the proposed algorithm, including how to design the cost function, how to construct and train the reconstruction model, and how to solve the cost function. Section 4 presents the results of our approach as well as comparing them with those produced by the peer methods. Finally, we conclude this paper with discussions on future work directions.

2. Motivation

Recently, expected patch log likelihood (EPLL) framework [23] using GMM prior for image denoising was proposed with its performance comparable to the state-of-the-art algorithms. EPLL trains a GMM using external natural image patches and, for each patch extracted from the unclean image, finds its highest probability solution in the model by the iterative splitting algorithm (for more details, the reader can refer to [23]). GMM prior shows its powerful denoising ability. However, this scheme is unsuitable for solving SISR problem, except for the time consuming during training phase, also because there is only one-scale image patches in the training data; thus it cannot introduce high frequency in the SR result and adopt the Wiener filter solution leading to damaging the fine details. We design a simple experiment to illustrate this point which was displayed in Figure 1. In this experiment, first, we upsample the small natural image “bear” with bicubic interpolation method as the input one (but adding no noise in it). Processing it with EPLL (the result in Figure 1(c)), we can notice that the salient edges are preserved very well, but the texture and details are almost erased. The result of EPLL is in agreement with our analysis. Processing the small natural image “bear” with our method, the result (Figure 1(d)) is sharper than the results of bicubic interpolation and EPLL methods. The algorithm of EPLL [23] is specialized for image denoising, which can preserve edge and erase the noise, and, however, it cannot add any correct high-frequency content in the results. So if the input is an image without high frequency, the result of EPLL algorithm is undesirable and our method can refine the result by finding some high frequency in the input image itself. Obviously, it is unfair for the specific denoising algorithm EPLL to be used for finding high frequency; we just use this experiment to illustrate the function of our algorithm.

Inspired by EPLL work, and also due to the image property of self-similarity between different scales, thus the input LR image and its filtering version can serve as a sufficient training data set for learning the model of the image scene. We propose a SISR method which trains GMM on the training data constructed by concatenating the different scales of the input image and can introduce the correct high-frequency component and protect the details in the HR result.

3. Our Approach

We first explain the notations appearing in our SISR scheme with the help of Figure 2. In Figure 2, the input image is denoted by $Y \in R^{\sqrt{n} \times \sqrt{n}}$, from which we obtain its low-frequency component image $Y_0 \in R^{\sqrt{n} \times \sqrt{n}}$ by low pass Gaussian

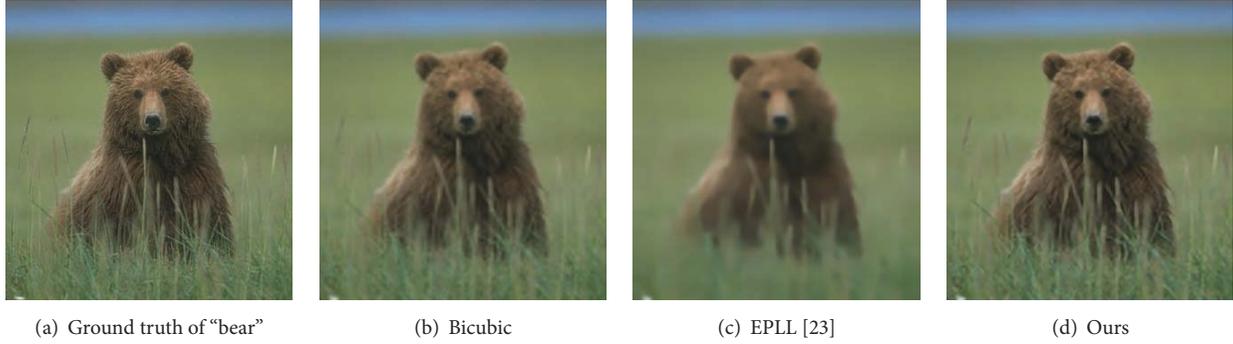


FIGURE 1: The comparison between the results of EPLL and the proposed method. (a) The ground truth image. Through downsampling and upsampling the image by the bicubic method, we derive (b), which is used as the input for producing (c) and (d) using the EPLL and the proposed method, respectively. From (c), we can see that EPLL fails to capture the image texture and its fine details. In comparison, the result of the proposed method (d) appears more natural and realistic; in particular the edges and textures regions are better preserved.

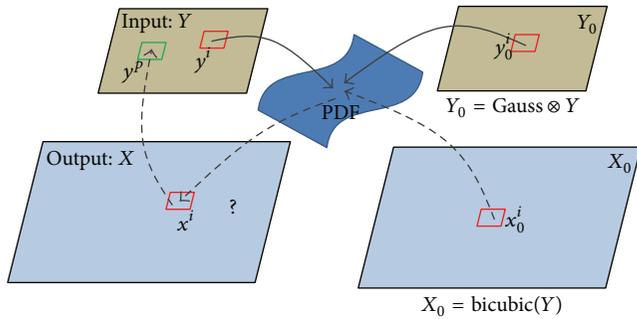


FIGURE 2: The framework of our approach. For a random patch x^i in X , we wish there exists at least one very similar patch y^p in Y . We find the joint PDF of the data pair $\{y_i, y_0^i\}_{i=1}^n$ and thus solve the optimal x^i when the data pair (x^i, x_0^i) substitute into the joint PDF.

filtering. We upsample Y using bicubic interpolation by a factor of s to get $X_0 \in R^{\sqrt{N} \times \sqrt{N}}$, $N = s^2 n$, and s is the upsampling factor. We use X_0 to approximate the low-frequency component version of the unknown high-resolution image $X \in R^{\sqrt{N} \times \sqrt{N}}$. From Y, Y_0 , and X_0 , the target is to estimate the HR image X . We use y^i and y_0^i to denote i th $a \times a$ image patch sampled from Y and Y_0 , respectively, and x^i and x_0^i to denote i th $a \times a$ image patch sampled from X and X_0 , respectively. $\{y_0^i\}_{i=1}^n$ and $\{x_0^i\}_{i=1}^N$ are thus referred to as the low-resolution image patches because they are missing high-frequency components, while $\{y^i\}_{i=1}^n$ and $\{x^i\}_{i=1}^N$ are referred to as their high-resolution counterparts separately. i represents the location of the patch in the corresponding images, and, in our scheme, the overlapped pixels between two neighbored patches are $(a-1)$; for the boundary pixels, we extend the image by symmetrical method. So the number of total patches extracted from one image is the same as the pixels of the image.

3.1. The Cost Function Based on Maximizing Self-Similarity Prior. For SISR problem, the generation of an LR image from the underlying HR image is expressed as

$$Y = DHX + w, \quad (1)$$

where $Y \in R^n$ is the observed LR image and $X \in R^N$ is the original HR image, $s = \sqrt{N/n}$ is the up-sampling factor, and $w \in R^n$ is assumed to be an additive Gaussian noise. Y, X , and w are all represented in vectorial form. $D \in R^{n \times N}$, $H \in R^{N \times N}$ are downsampling and filtering matrix, respectively.

For a random patch x^i in X , Freeman et al. [11, 15], Chang et al. [17], Yang et al. [14], and Dong et al. [12] all want to find one nearest neighbor patch y^p in Y , by approximating the distance between x^i in X and y^p in Y to the distance between x_0^i in X_0 and y_0^p in Y_0 . However, the approximation is not accurate, since the correspondence between the LR image patches and their counterparts (HR image patches) and the neighborhood relationship cannot be preserved perfectly due to the ‘‘one-to-many’’ mapping existing between one LR image and many HR images.

If we have already got the reconstructed HR image X , according to the self-similarity property, a random patch x^i in X should at least have one nearest neighbor patch y^p in Y , and for another random patch x^j in X , being not afraid of the overlapping situation in the low resolution image Y , it should also at least have one nearest neighbor patch y^q in Y . We use $p(P_i X)$ to measure the similarity between x^i and y^p . P_i is a matrix to extract the patch x^i from X . The larger the probability value $p(P_i X)$, the more similar the appearance between x^i and y^p . For any patch x^i extracted from the reconstructed HR image, we hope the probability $p(x^i)$ of the patch obtains the maximizing value. And we call the property of the reconstructed HR image Maximizing Self-Similarity Prior (MSSP). So, for the whole reconstructed HR image, we can write out the cost function as

$$\arg \min_X \|DHX - Y\|_2^2 - \lambda \prod_{i=1}^N p(P_i X), \quad (2)$$

where λ is the parameter for balancing the effect of the fidelity term $\|DHX - Y\|_2^2$ and the regularization prior $\prod_{i=1}^N p(P_i X)$.

3.2. Solving. Since there are so many multiple operations in the second term of (2), it is difficult to solve, and $p(P_i X)$ represents probability which is always positive real value; thus

we can use log function with the second term in this equation, and the cost function is changed as follows:

$$\arg \min_X \|DHX - Y\|_2^2 - \lambda \sum_{i=1}^N \log p(P_i X). \quad (3)$$

The cost function is nonconvex and difficult to solve. For finding the optimization solution, we introduce a set of auxiliary variables $\{z_i\}_{i=1}^N$, the overlapped patch $P_i X$ corresponding to z_i . Thus the cost function is changed to

$$\begin{aligned} \arg \min_{X, \{z_i\}_{i=1}^N} & \|DHX - Y\|_2^2 - \lambda \sum_{i=1}^N \log p(z_i) \\ & + \beta \sum_{j=1}^N \|P_j X - z_j\|_2^2. \end{aligned} \quad (4)$$

Note that when $\beta \rightarrow +\infty$, since we want to find the minimize solution of (4), $P_j X$ is equal to z_j , and the solution of the equation will be converged. We solve this equation by taking an iterative way. Firstly, while we keep z_i fixed, solve for X . This can be solved by setting the derivative of this equation about X to zero and we can get the result

$$\begin{aligned} \widehat{X} = & \left(H^T D^T D H + \beta \sum_{j=1}^N P_j^T P_j \right)^{-1} \\ & \cdot \left(H^T D^T Y + \beta \sum_{j=1}^N P_j^T z_j \right), \end{aligned} \quad (5)$$

where $(A)^T$ represents transposing matrix A , $(A)^{-1}$ solves the inverse matrix of A , and j is for all overlapping patches in X . And then solving for $\{z_i\}_{i=1}^N$ gives \widehat{X} . After each iteration, we increase β and continue to do the next iteration. In next two following subsections, we will elaborate how to solve $\{z_i\}_{i=1}^N$.

3.2.1. Training. For solving each auxiliary variable z_i , it just means estimating the most likely image patch under prior of MSSP. However, we have no HR image X in practice; thus it is difficult to resolve directly. Here, we consider the relationship between Y and Y_0 and extract patches $\{y^p\}_{p=1}^n$ and $\{y_0^p\}_{p=1}^n$ from the input image Y and the Gaussian filtering version Y_0 separately. For representing patch textures and details rather than absolute intensities, we subtract the mean value for each patch. And then constructing the training data set D , in which each element $d^p \in R^{2a^2}$ is formed by concatenating y^p and y_0^p in vectorial form,

$$d^p = \begin{bmatrix} y_0^p \\ y^p \end{bmatrix}, \quad p = 1, \dots, N. \quad (6)$$

Gaussian mixture model (GMM) is a parametric probability density function (PDF) that is represented as a weighted sum of Gaussian densities. GMM can approximate an arbitrary probability density function accurately [24]. For

a random variable $d \in R^{2a^2}$, we learn a GMM over these concatenated vectors and write its PDF by GMM as

$$p(d) = \sum_{k=1}^K \alpha_k N(d | \mu_k, \Sigma_k), \quad (7)$$

where K is the number of Gaussian mixture components and α_k , μ_k , and Σ_k are the parameters of weight, mean vector, and covariance matrix of k th Gaussian mixture component separately. Using EM algorithm to estimate these parameters is not hard.

3.2.2. Predicting. We consider the parameters $\{\alpha_k, \mu_k, \Sigma_k\}$ and can rewrite $\mu_k = [\mu_k^{y_0}, \mu_k^y]^T$, where $\mu_k^{y_0}$ and μ_k^y are the mean vectors of image patches $\{y_0^p\}_{p=1}^N$ and $\{y^p\}_{p=1}^N$ separately; let us also introduce the notation $\Sigma_k = \begin{bmatrix} \Sigma_k^{y_0} & \Sigma_k^{y_0 y} \\ \Sigma_k^{y_0 y} & \Sigma_k^y \end{bmatrix}$ for the covariance matrix of the model component densities. According to the self-similarity between the different scales of one image, for a random patch z^i in the reconstructed HR image, we can predict the occurrence probability in the original input image as

$$p(z^i | x_0^i, \{\alpha_k, \mu_k, \Sigma_k\}_{k=1}^K) = N(z^i | \widehat{\mu}, \widehat{\Sigma}), \quad (8)$$

where

$$\begin{aligned} \widehat{\mu} &= \sum_{k=1}^K \varphi_k(x_0^i) \left[\mu_k^y + \Sigma_k^{y y_0} (\Sigma_k^{y_0})^{-1} (x_0^i - \mu_k^{y_0}) \right], \\ \widehat{\Sigma} &= \sum_{k=1}^K \varphi_k^2(x_0^i) \left[\Sigma_k^y + \Sigma_k^{y y_0} (\Sigma_k^{y_0})^{-1} \Sigma_k^{y_0 y} \right], \end{aligned} \quad (9)$$

$$\varphi_k(x_0^i) = \frac{\alpha_k N(x_0^i | \mu_k^{y_0}, \Sigma_k^{y_0})}{\sum_{k'=1}^K \alpha_{k'} N(x_0^i | \mu_{k'}^{y_0}, \Sigma_{k'}^{y_0})}.$$

Based on (3), prediction can be obtained by taking the expectation result:

$$z^i = E(z^i | x_0^i, \{\alpha_k, \mu_k, \Sigma_k\}_{k=1}^K) = \widehat{\mu}. \quad (10)$$

At last, the algorithm for SISR using maximizing self-similarity prior is summarized as in Algorithm 1.

4. Experiments and Analysis

4.1. Parameter Setting. Experiments have been conducted to evaluate the result of our method in comparison with several state-of-the-art algorithms. We start by using the original image as LR input and upsample it with a scale factor of 2. For further upsampling, we used the previous output as the input and solved its HR image. Note that, for a color image, it is first transformed from RGB to YIQ. Then, the Y channel (intensity) is upsampled by our algorithm because human vision is more sensitive to brightness change. I and Q channels are interpolated to the desired size by the bicubic interpolation method. Finally, the three channels

```

Input: LR image  $Y$ 
(1)  $X_0 = \text{Bicubic}(Y)$ ;
(2)  $Y_0 = \text{Gaussian} \otimes Y$ 
(3) Extract patches  $\{y^p\}_{p=1}^n$  and  $\{y_0^p\}_{p=1}^n, \{x_0^i\}_{i=1}^N$ ;
(4) construct training data  $d^p = [y_0^p, y^p]^T$ ;
(5) train GMM;
(6) initialize  $X$  with  $X_0$ ;
(7) for  $i = 1 : t$  //  $t$  represents the number of the iterative times
    Solve  $X$  with (5), keeping  $\{z^i\}_{i=1}^N$  fixed;
    Update  $X$ ;
    Solve  $\{z^i\}_{i=1}^N$  with (10), keeping  $X$  fixed;
    Update  $\{z^i\}_{i=1}^N$ ;
end
Output: HR reconstructed Image  $X$ 

```

ALGORITHM 1: SISR using MSSP.

are combined to form the final HR result. The input LR image Y is generated from the original HR image by a down-sampling and filtering process, the low-frequency band X_0 of the unknown HR image is approximated by bicubic interpolation from Y , and the low-frequency band of the input LR image Y_0 is obtained by low pass Gaussian filtering with a standard deviation of 0.5. In all experiments, the size of all patches is 5×5 ($a = 5$), $\lambda = 0.5$, $\beta = (t \times 10)^2$, and t represents the t th iteration. T represents the number of the iterative times; in this paper, we set $T = 5$. As we know, the amount of GMM component densities K is important to the performance of the joint PDF learning. However, the accuracy and the efficiency should be trade-off in practical implementation. In our experiments, we apply a Bayesian-based model selection criterion [25] to determine the number of GMM components by minimizing the following metric:

$$K = -2 \sum_{i=1}^N \log \left(p \left(\{d^p\}_{p=1}^N \mid \{\alpha_k, \mu_k, \Sigma_k\} \right) \right) + \delta \log N, \quad (11)$$

where δ is the total number of model parameters.

4.2. Visual Analysis. Subjective visual effect is a significant evaluation to superresolution models, which could reflect the advantages and disadvantages intuitively. In this section, we test our model and make some comparisons to the state-of-the-art models.

As mentioned before, Glasner et al. [20] emphasized the importance of image self-similarity to superresolution problem. Therefore, in Figure 3, we present a comparison with respect to Glasner's patch recurrence (PR) model, our model, and the ground truth. As observed in Figure 3(b), Glasner's patch has some artifacts along the sharp edges and tends to be smoother. By contrast, our model could obtain an output closer to the ground truth. The difference between both of the results could be observed clearly by the close-up and its corresponding pseudocolor version.

In Figure 4, we adopt the natural images which have obvious self-similarity texture and structure components to

evaluate our model. For objectiveness, we select the Gaussian-type model to compare. Figure 4 presents the results which produced by He's Gaussian process regression (GPR) model [22] and ours. In the texture case, we note that our result has better detail performance rather than blurring. And in the structure case, we could see that some artifacts appear in GPR result but not in ours.

Furthermore, in 2x upsampling, we compare between Shan's method [6], Glasner's patch recurrence method [20], Yang's SCSR method [13], Peleg's statistical prediction model [26], and ours. The tested images include such characteristics of sharp edges, complicated textures, and repeated structures. In Figure 5, by the close-up, we observe that Shan's, Glasner's, and Peleg's results would yield some artifacts along the sharp edges, for example, the wheel image in the second line. And Yang's SCSR model would generate a relative blurred result. Generally speaking, our model could obtain a series of satisfactory results in these conventional tested images.

4.3. Objective Evaluation. Peak single-to-noise ratio (PSNR) and structure similarity (SSIM) index are used as the objective measures. PSNR is defined as

$$\text{PSNR} = 10 \log_{10} \frac{m \times 255^2}{\sum_i \hat{x}_i - x_i}, \quad (12)$$

where x is the original HR image, \hat{x} is the SR result we approximated, and m is the total number of pixels in the image.

SSIM is more consistent with human eye than PSNR. The SSIM metric is calculated on various windows of an image and the measure between two windows x and y of common size $N \times N$ is

$$\text{SSIM}(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_x \sigma_y + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (13)$$

where μ_x, μ_y are the average values of x and y , σ_x^2, σ_y^2 are the variances of x and y , and c_1, c_2 are two variables to stabilize the division with weak denominator. From the definition, we can

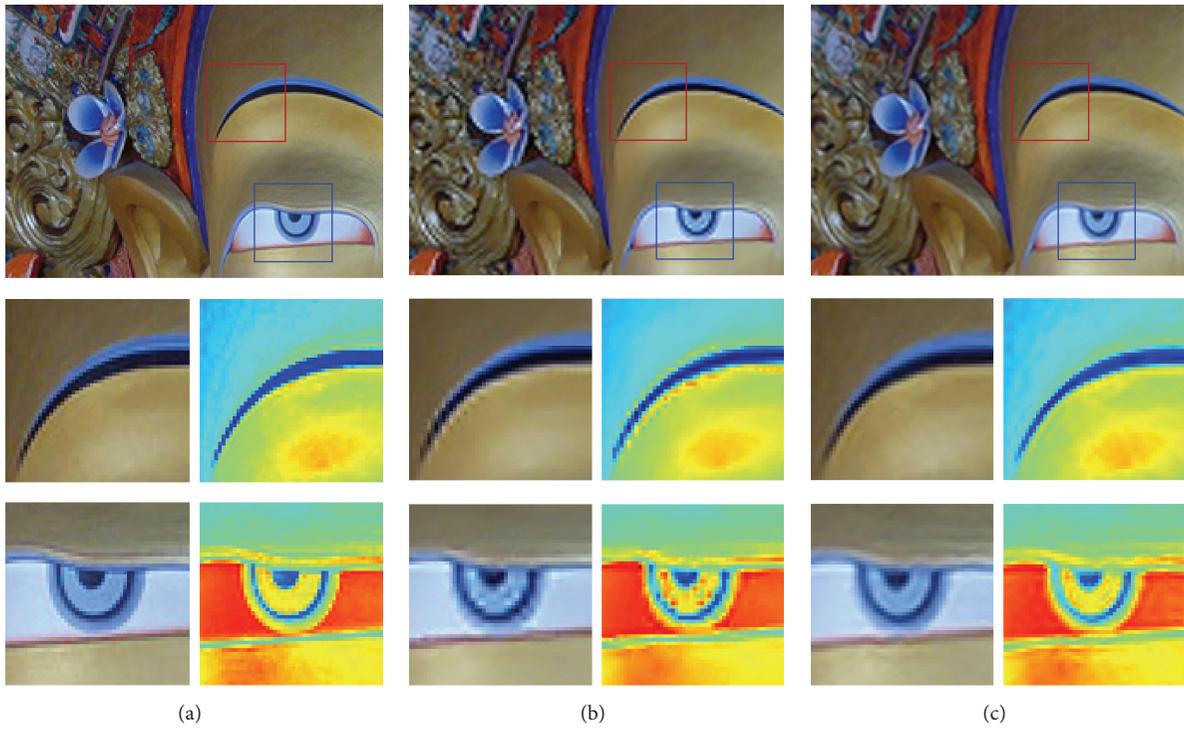


FIGURE 3: Comparisons with the self-similarity measurement. (a) Ground truth. (b) Glasner's PR [20]. (c) Our proposed method. Note the artifacts in (b) and ours are very close to the ground truth.

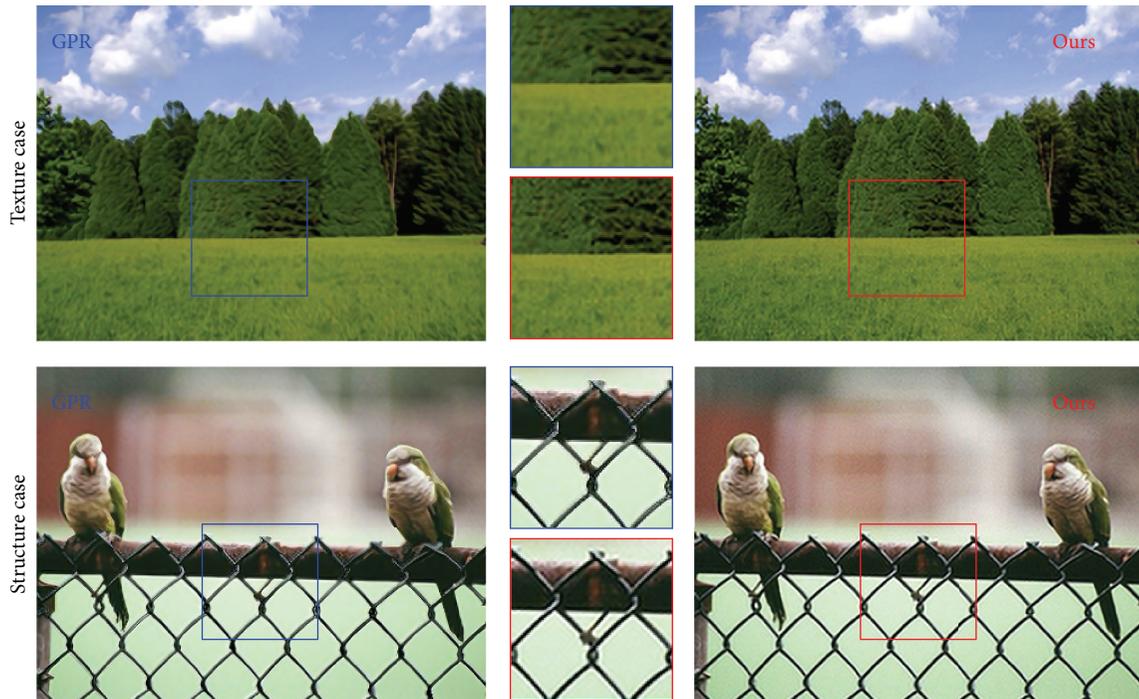


FIGURE 4: Texture and structure cases analysis. He's Gaussian process regression model [22] is compared with ours.

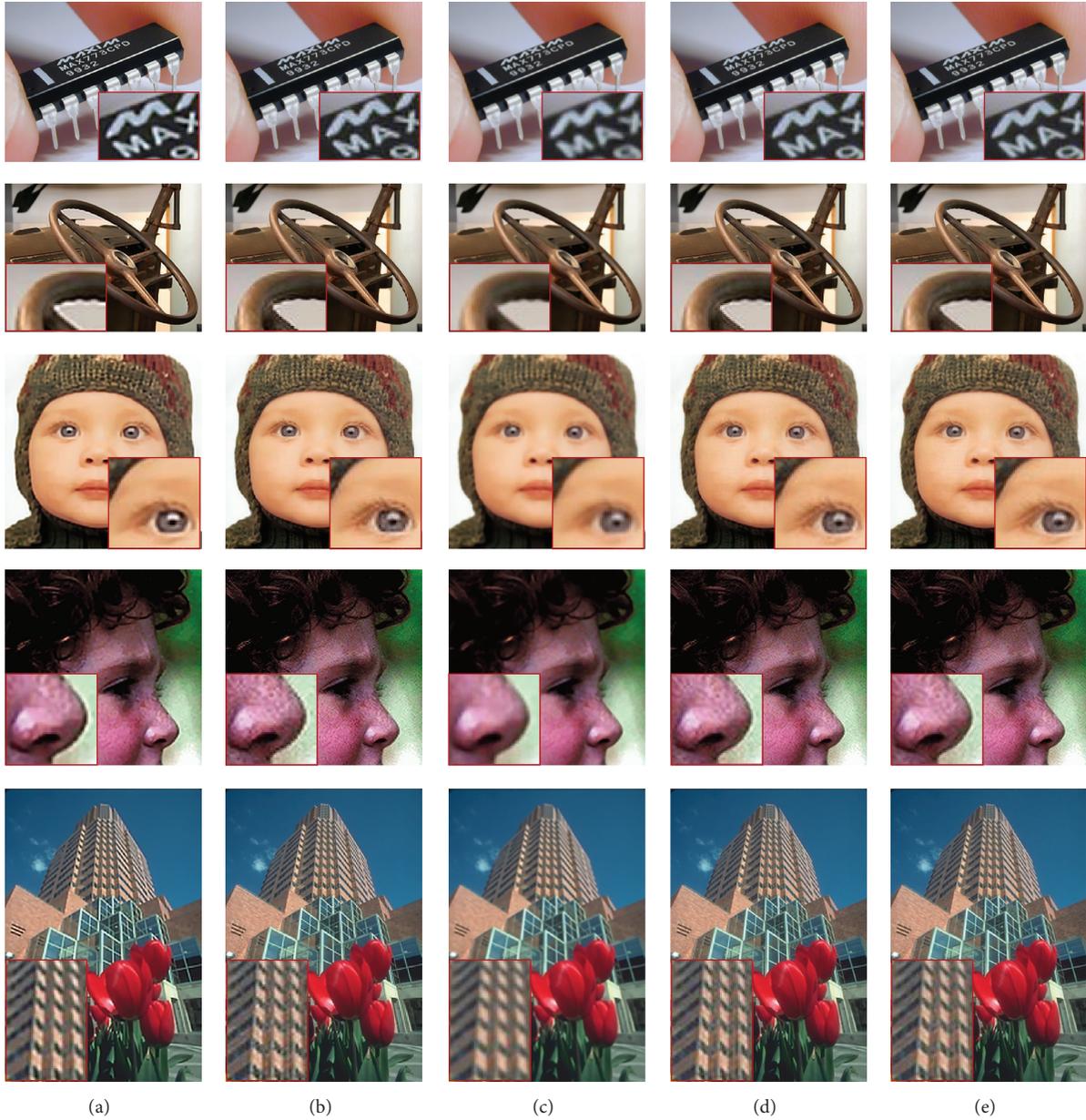


FIGURE 5: Comparisons with 2x magnification. (a) Shan’s method [6]. (b) Glasner’s patch recurrence method [20]. (c) Yang’s SCSR [13]. (d) Peleg’s SPM [26]. (e) Ours.

see that SR results with better quality should provide larger PSNR and SSIM values.

In our experiments, fifteen images are randomly selected from RetargetMe benchmark (<http://people.csail.mit.edu/mrub/retargetme/>), as illustrated in Figure 6. We first apply the Lanczos resampling to yield the low-resolution version (0.5x) and then recover to the original size to compare with the ground truth. Five state-of-the-art models are tested, including Glasner’s patch recurrence method [20], SCSR [13], KRR [27], GPR [22], and SPM [26]. The PSNR and SSIM metrics are used to evaluate these results, and the corresponding scores are, respectively, recorded in Table 1. Note

that our method could obtain the best or the second-best performance in most of the test images.

For saving the computing time, we can compute $(H^T D^T D H + \beta \cdot \sum_{k=1}^N P_i^T P_i)^{-1}$, $(D H)^T$ previously, compute $\Sigma_k^{y_0} (\Sigma_k^{y_0})^{-1}$, $\varphi_k(x_0^i)$, and $(D H)^T Y$ after finishing the GMM training phrase, and use it directly in the iterative process. Figure 7 provides the numerical comparison on executing time with the state-of-the-art algorithms. The images are upsampled from 256×256 to 768×768 with the conventional images. The hardware configuration of our modern computer is Intel CPU 1.7 GHZ, 4 GB memory, and the software platform is Matlab 7.1.



FIGURE 6: Fifteen test images for effectiveness validation. From left to right: bells, bike, Buddha, butterfly, child, eagle, face, family, Fatem, fish, surfer, tiger, trees, two birds, and waterfall. All the test image are extracted from RetargetMe data set. And the validation data are recorded in Table 1.

TABLE 1: PSNR and SSIM for scale factor of 2.

| | PR [20] | SCSR [13] | KRR [27] | GPR [22] | SPM [26] | Ours |
|-----------|---------|-----------|---------------|----------|---------------|---------------|
| Bells | 25.401 | 29.477 | 29.527 | 28.213 | 29.568 | 29.511 |
| | 0.838 | 0.904 | 0.904 | 0.889 | 0.907 | 0.905 |
| Bike | 21.602 | 25.801 | 25.801 | 24.110 | 25.803 | 25.884 |
| | 0.713 | 0.845 | 0.847 | 0.779 | 0.846 | 0.848 |
| Buddha | 28.170 | 32.033 | 32.187 | 28.338 | 32.168 | 32.194 |
| | 0.851 | 0.915 | 0.919 | 0.844 | 0.918 | 0.916 |
| Butterfly | 27.337 | 31.743 | 32.128 | 26.352 | 31.883 | 31.983 |
| | 0.940 | 0.969 | 0.974 | 0.931 | 0.97 | 0.973 |
| Child | 26.435 | 29.941 | 30.154 | 26.392 | 30.021 | 30.248 |
| | 0.893 | 0.941 | 0.945 | 0.882 | 0.943 | 0.944 |
| Eagle | 27.715 | 30.293 | 30.676 | 27.740 | 30.280 | 30.339 |
| | 0.848 | 0.909 | 0.911 | 0.837 | 0.912 | 0.908 |
| Face | 30.307 | 37.054 | 36.953 | 30.453 | 37.209 | 37.381 |
| | 0.920 | 0.967 | 0.970 | 0.920 | 0.971 | 0.970 |
| Family | 25.767 | 30.307 | 30.295 | 25.972 | 30.443 | 30.465 |
| | 0.803 | 0.908 | 0.907 | 0.793 | 0.910 | 0.911 |
| Fatem | 29.091 | 35.384 | 35.620 | 28.701 | 35.788 | 35.857 |
| | 0.882 | 0.953 | 0.956 | 0.875 | 0.957 | 0.956 |
| Fish | 27.304 | 34.609 | 35.160 | 27.204 | 34.372 | 34.934 |
| | 0.859 | 0.950 | 0.954 | 0.860 | 0.950 | 0.952 |
| Surfer | 25.294 | 28.178 | 28.199 | 25.357 | 28.176 | 28.236 |
| | 0.736 | 0.820 | 0.820 | 0.712 | 0.818 | 0.822 |
| Tiger | 23.727 | 29.103 | 29.143 | 23.643 | 29.248 | 29.321 |
| | 0.883 | 0.946 | 0.948 | 0.880 | 0.949 | 0.948 |
| Trees | 27.848 | 30.678 | 30.651 | 27.928 | 30.794 | 30.780 |
| | 0.734 | 0.851 | 0.850 | 0.752 | 0.850 | 0.854 |
| Two birds | 21.112 | 26.262 | 27.024 | 21.525 | 26.309 | 26.487 |
| | 0.687 | 0.756 | 0.767 | 0.688 | 0.756 | 0.757 |
| Waterfall | 23.876 | 26.329 | 26.421 | 25.201 | 26.390 | 26.360 |
| | 0.771 | 0.838 | 0.841 | 0.794 | 0.840 | 0.840 |

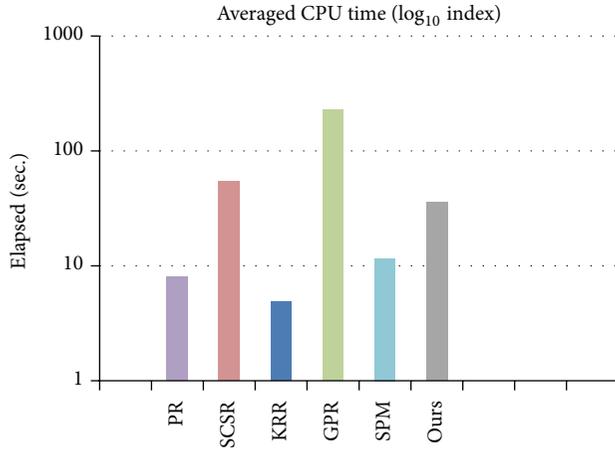


FIGURE 7: Comparison of the averaged CPU time between different methods.

4.4. Analysis. By analyzing (8) and (10), we can notice that z_i is decomposed into K layers, and, for each layer, the weight is $\varphi_k(x_0^i)$. This operation can avoid the influence of different frequencies. In each layer, we can see that if $\Sigma_k^{y_0}(\Sigma_k^{y_0})^{-1} = I$, the result of z_i is degenerated into the result of the bicubic interpolation method. In fact, we can consider $\Sigma_k^{y_0}(\Sigma_k^{y_0})^{-1}$ as transforming matrix in which $\Sigma_k^{y_0}(\Sigma_k^{y_0})^{-1}y_0^i = y^i$. We can assume that the relationship between $\{y_0^i\}_{i=1}^n$ and $\{y^i\}_{i=1}^n$ is similar to the relationship between $\{x_0^i\}_{i=1}^N$ and $\{x^i\}_{i=1}^N$. Thus $x^i \approx \Sigma_k^{y_0}(\Sigma_k^{y_0})^{-1}x_0^i$. So our approach is often superior to those generated by other state-of-the-art algorithms.

5. Conclusion

In this paper, a novel algorithm is proposed for SISR based on the concept of image self-similarity maximization. The proposed method carefully observes and leverages a pair of local and global image constraints between different resolutions of an image. The local constraint dictates that a random patch in the reconstructed HR image should exhibit visual similarity with its corresponding image region in the LR image; the global constraints state that the reconstructed HR image should perceptually resemble the input LR image through downsampling and filtering. The comparative experimental results show that the results generated by the new method preserve details of image more realistically, as determined both perceptually through subjective evaluation and quantitatively via numeric measurements. In the future, we plan to adapt and migrate the method for the problem of video superresolution.

Conflict of Interests

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

Acknowledgments

This research is supported by the National Natural Science Foundation of China (61320106008, 61262050, and

61363049), NSFC-Guangdong Joint Fund (no. U1135003), Guangdong Province Special Fund Research Project for Scientific and Professional Construction of University (2013WYXM0058), and Guangdong Province Science and Technology Planning Project (2013B020314019).

References

- [1] H. S. Hou and H. C. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, no. 6, pp. 508–517, 1978.
- [2] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1521–1527, 2001.
- [3] L. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2226–2238, 2006.
- [4] M. Li and T. Q. Nguyen, "Markov random field model-based edge-directed image interpolation," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1121–1128, 2008.
- [5] R. Fattal, "Image upsampling via imposed edge statistics," *ACM Transactions on Graphics*, vol. 26, no. 3, article 95, 2007.
- [6] Q. Shan, Z. Li, J. Jia, and C.-K. Tang, "Fast image/video upsampling," *ACM Transactions on Graphics*, vol. 27, no. 5, article 153, 2008.
- [7] J. Sun, J. Sun, Z. Xu, and H.-Y. Shum, "Image super-resolution using gradient profile prior," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, Anchorage, Alaska, USA, June 2008.
- [8] C. Chen, H. Liang, S. Zhao, Z. Lyu, and M. Sarem, "A novel multi-image super-resolution reconstruction method using anisotropic fractional order adaptive norm," *The Visual Computer*, vol. 31, no. 9, pp. 1217–1231, 2015.
- [9] X. Lu, Y. Sun, and Y. Yuan, "Image reconstruction by an alternating minimisation," *Neurocomputing*, vol. 74, no. 5, pp. 661–670, 2011.
- [10] B. Shi, H. Zhao, M. Ben-Ezra et al., "Sub-pixel layout for super-resolution with images in the octic group," in *Proceedings of the European Conference on Computer Vision*, pp. 250–264, Zurich, Switzerland, September 2014.
- [11] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.
- [12] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part IV*, vol. 8692 of *Lecture Notes in Computer Science*, pp. 184–199, Springer, 2014.
- [13] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3467–3478, 2012.
- [14] J. Yang, Z. Lin, and S. Cohen, "Fast image super-resolution based on in-place example regression," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 1059–1066, Portland, Ore, USA, June 2013.
- [15] W. T. Freeman and C. Liu, "Markov random fields for super-resolution and texture synthesis," in *Advances in Markov Random Fields for Vision and Image Processing*, The MIT Press, 2011.
- [16] M.-T. Chi, C.-Y. Yao, E. Zhang, and T.-Y. Lee, "Optical illusion shape texturing using repeated asymmetric patterns," *The Visual Computer*, vol. 30, no. 6–8, pp. 809–819, 2014.

- [17] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, pp. 1275–1282, Washington, DC, USA, July 2004.
- [18] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, IEEE, Anchorage, Alaska, USA, June 2008.
- [19] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.
- [20] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *Proceedings of the 12th IEEE International Conference on Computer Vision (ICCV '09)*, pp. 349–356, IEEE, Kyoto, Japan, September-October 2009.
- [21] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Transactions on Graphics*, vol. 30, no. 2, article 12, 11 pages, 2011.
- [22] H. He and W.-C. Siu, "Single image super-resolution using Gaussian process regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 449–456, IEEE, Providence, RI, USA, June 2011.
- [23] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 479–486, IEEE, Barcelona, Spain, November 2011.
- [24] J. A. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," *International Computer Science Institute*, vol. 4, no. 510, pp. 2–7, 1998.
- [25] G. Schwarz, "Estimating the dimension of a model," *Institute of Mathematical Statistics*, vol. 6, no. 2, pp. 461–464, 1978.
- [26] T. Peleg and M. Elad, "A statistical prediction model based on sparse representations for single image super-resolution," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2569–2582, 2014.
- [27] K. I. Kim and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1127–1133, 2010.