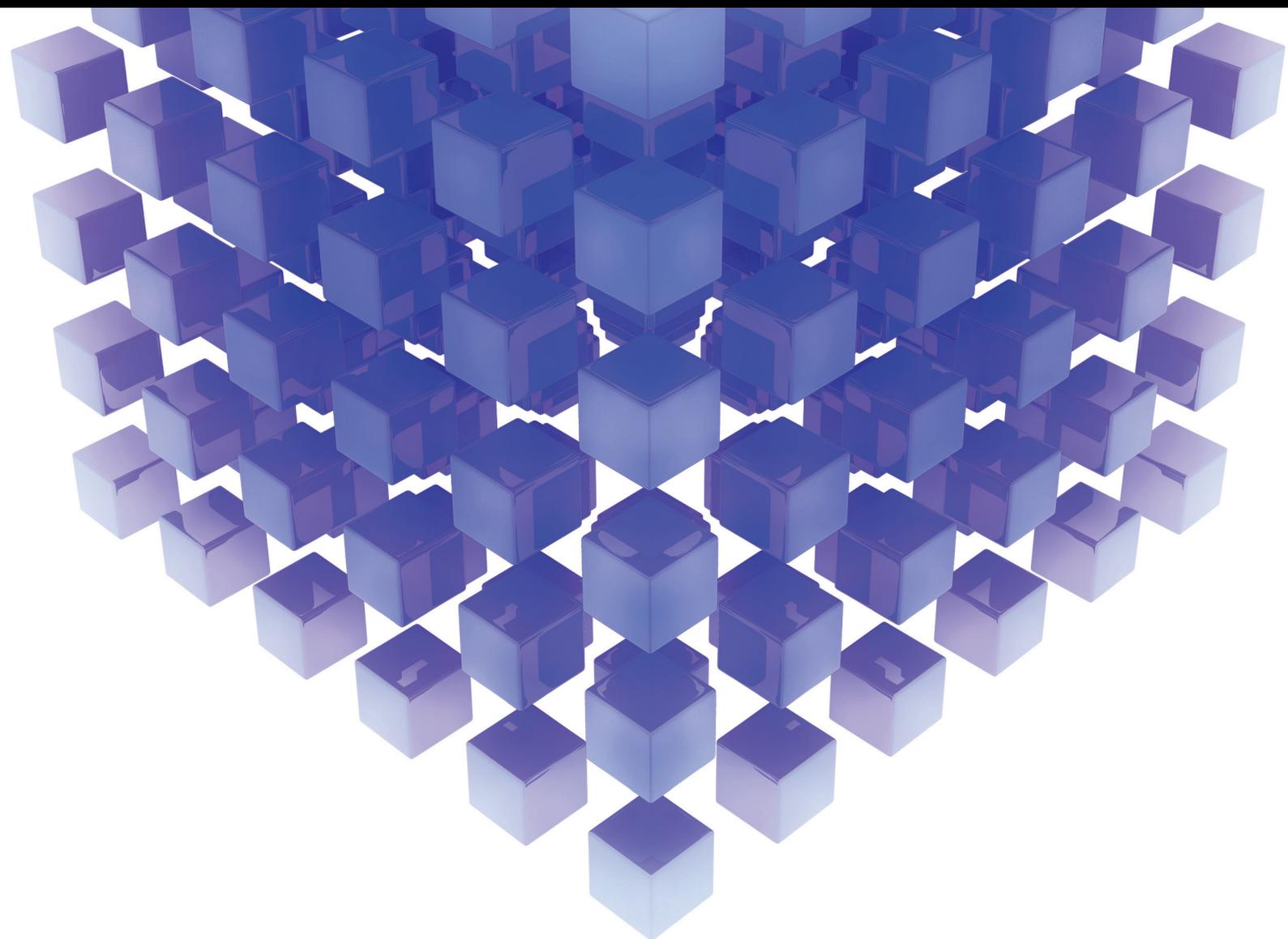


Mathematical Problems in Engineering

# Advances in Variational and Partial Differential Equation-Based Models for Image Processing and Computer Vision

Lead Guest Editor: Tudor Barbu

Guest Editors: Gabriela Marinoschi, Costică Moroşanu, and Ionuţ Munteanu





---

**Advances in Variational and Partial Differential  
Equation-Based Models for Image Processing  
and Computer Vision**

Mathematical Problems in Engineering

---

# **Advances in Variational and Partial Differential Equation-Based Models for Image Processing and Computer Vision**

Lead Guest Editor: Tudor Barbu

Guest Editors: Gabriela Marinoschi, Costică Moroşanu,  
and Ionuţ Munteanu



---

Copyright © 2018 Hindawi. 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  
AITOUCHE Abdelouhab, France  
Leonardo Acho, Spain  
José A. Acosta, Spain  
Paolo Adesso, Italy  
Claudia Adduce, Italy  
Ramesh Agarwal, USA  
Francesco Aggogeri, Italy  
Juan C. Agüero, Australia  
R Aguilar-López, Mexico  
Tarek Ahmed-Ali, France  
Elias Aifantis, USA  
Muhammad N. Akram, Norway  
Guido Ala, Italy  
Andrea Alaimo, Italy  
Reza Alam, USA  
Nicholas Alexander, UK  
Salvatore Alfonzetti, Italy  
Mohammad D. Aliyu, Canada  
Juan A. Almendral, Spain  
José Domingo Álvarez, Spain  
Cláudio Alves, Portugal  
J. P. Amezquita-Sanchez, Mexico  
Lionel Amodeo, France  
Sebastian Anita, Romania  
Renata Archetti, Italy  
Felice Arena, Italy  
Sabri Arik, Turkey  
Francesco Aristodemo, Italy  
Fausto Arpino, Italy  
Alessandro Arsie, USA  
Edoardo Artioli, Italy  
Fumihiko Ashida, Japan  
Farhad Aslani, Australia  
Mohsen Asle Zaem, USA  
Romain Aubry, USA  
Matteo Aureli, USA  
Richard I. Avery, USA  
Viktor Avrutin, Germany  
Francesco Aymerich, Italy  
Sajad Azizi, Belgium  
Michele Baccocchi, Italy  
Seungik Baek, USA  
Adil Bagirov, Australia  
Khaled Bahlali, France  
Laurent Bako, France  
Pedro Balaguer, Spain  
Stefan Balint, Romania  
Ines Tejado Balsera, Spain  
Alfonso Banos, Spain  
Jerzy Baranowski, Poland  
Roberto Baratti, Italy  
Andrzej Bartoszewicz, Poland  
David Bassir, France  
Chiara Bedon, Italy  
Azeddine Beghdadi, France  
Denis Benasciutti, Italy  
Ivano Benedetti, Italy  
Rosa M. Benito, Spain  
Elena Benvenuti, Italy  
Giovanni Berselli, Italy  
Michele Betti, Italy  
Jean-Charles Beugnot, France  
Pietro Bia, Italy  
Carlo Bianca, France  
Simone Bianco, Italy  
Vincenzo Bianco, Italy  
Vittorio Bianco, Italy  
Gennaro N. Bifulco, Italy  
David Bigaud, France  
Antonio Bilotta, Italy  
Paul Bogdan, USA  
Guido Bolognesi, UK  
Rodolfo Bontempo, Italy  
Alberto Borboni, Italy  
Paolo Boscaroli, Italy  
Daniela Boso, Italy  
Guillermo Botella-Juan, Spain  
Boulaïd Boulkroune, Belgium  
Fabio Bovenga, Italy  
Francesco Braghin, Italy  
Ricardo Branco, Portugal  
Maurizio Brocchini, Italy  
Julien Bruchon, France  
Matteo Bruggi, Italy  
Michele Brun, Italy  
Vasilis Burganos, Greece  
Tito Busani, USA  
Raquel Caballero-Águila, Spain  
Filippo Cacace, Italy  
Pierfrancesco Cacciola, UK  
Salvatore Caddemi, Italy  
Roberto Caldelli, Italy  
Alberto Campagnolo, Italy  
Eric Campos-Canton, Mexico  
Marko Canadija, Croatia  
Salvatore Cannella, Italy  
Francesco Cannizzaro, Italy  
Javier Cara, Spain  
Ana Carpio, Spain  
Caterina Casavola, Italy  
Sara Casciati, Italy  
Federica Caselli, Italy  
Carmen Castillo, Spain  
Inmaculada T. Castro, Spain  
Miguel Castro, Portugal  
Giuseppe Catalanotti, UK  
Nicola Caterino, Italy  
Alberto Cavallo, Italy  
Gabriele Cazzulani, Italy  
Luis Cea, Spain  
Miguel Cerrolaza, Venezuela  
M. Chadli, France  
Gregory Chagnon, France  
Ludovic Chamoin, France  
Ching-Ter Chang, Taiwan  
Qing Chang, USA  
Michael J. Chappell, UK  
Kacem Chehdi, France  
Peter N. Cheimets, USA  
Xinkai Chen, Japan  
Luca Chiapponi, Italy  
Francisco Chicano, Spain  
Nicholas Chileshe, Australia  
Adrian Chmielewski, Poland  
Ioannis T. Christou, Greece  
Hung-Yuan Chung, Taiwan  
Simone Cinquemani, Italy  
Roberto G. Citarella, Italy  
Joaquim Ciurana, Spain  
John D. Clayton, USA  
Francesco Clementi, Italy

Piero Colajanni, Italy  
Giuseppina Colicchio, Italy  
Vassilios Constantoudis, Greece  
Enrico Conte, Italy  
Francesco Conte, Italy  
Alessandro Contento, USA  
Mario Cools, Belgium  
José A. Correia, Portugal  
Jean-Pierre Corriou, France  
J.-C. Cortés, Spain  
Carlo Cosentino, Italy  
Paolo Crippa, Italy  
Andrea Crivellini, Italy  
Frederico R. B. Cruz, Brazil  
Erik Cuevas, Mexico  
Maria C. Cunha, Portugal  
Peter Dabnichki, Australia  
Luca D'Acierno, Italy  
Weizhong Dai, USA  
Andrea Dall'Asta, Italy  
Purushothaman Damodaran, USA  
Farhang Daneshmand, Canada  
Giuseppe D'Aniello, Italy  
Sergey N. Dashkovskiy, Germany  
Fabio De Angelis, Italy  
Samuele De Bartolo, Italy  
Pietro De Lellis, Italy  
Luciano R. O. de Lima, Brazil  
Alessandro De Luca, Italy  
Stefano de Miranda, Italy  
Filippo de Monte, Italy  
M. do Rosário de Pinho, Portugal  
Michael Defoort, France  
Alessandro Della Corte, Italy  
Xavier Delorme, France  
Laurent Dewasme, Belgium  
Angelo Di Egidio, Italy  
Roberta Di Pace, Italy  
Ramón I. Diego, Spain  
Yannis Dimakopoulos, Greece  
Zhengtao Ding, UK  
M. Djemai, France  
Alexandre B. Dolgui, France  
Georgios Dounias, Greece  
Florent Duchaine, France  
George S. Dulikravich, USA  
Bogdan Dumitrescu, Romania  
Horst Ecker, Austria  
Saeed Eftekhari Azam, USA  
Ahmed El Hajjaji, France  
Antonio Elipse, Spain  
Fouad Erchiqui, Canada  
Anders Eriksson, Sweden  
R. Emre Erkmen, Australia  
Gilberto Espinosa-Paredes, Italy  
Andrea L. Facci, Italy  
Giacomo Falcucci, Italy  
Giovanni Falsone, Italy  
Hua Fan, China  
Nicholas Fantuzzi, Italy  
Yann Favennec, France  
Fiorenzo A. Fazzolari, UK  
Giuseppe Fedele, Italy  
Roberto Fedele, Italy  
Arturo J. Fernández, Spain  
Jesus M. Fernandez Oro, Spain  
Massimiliano Ferrara, Italy  
Francesco Ferrise, Italy  
Eric Feulvarch, France  
Barak Fishbain, Israel  
S. Douwe Flapper, Netherlands  
Thierry Floquet, France  
Eric Florentin, France  
Alessandro Formisano, Italy  
Francesco Franco, Italy  
Elisa Francomano, Italy  
Tomonari Furukawa, USA  
Juan C. G. Prada, Spain  
Mohamed Gadala, Canada  
Matteo Gaeta, Italy  
Mauro Gaggero, Italy  
Zoran Gajic, USA  
Erez Gal, Israel  
Jaime Gallardo-Alvarado, Mexico  
Ugo Galvanetto, Italy  
Akemi Gálvez, Spain  
Rita Gamberini, Italy  
Maria L. Gandarias, Spain  
Arman Ganji, Canada  
Zhiwei Gao, UK  
Zhong-Ke Gao, China  
Giovanni Garcea, Italy  
Luis Rodolfo Garcia Carrillo, USA  
Jose M. Garcia-Aznar, Spain  
Akhil Garg, China  
Alessandro Gasparetto, Italy  
Oleg V. Gendelman, Israel  
Stylios Georgantzinos, Greece  
Fotios Georgiades, UK  
Mergen H. Ghayesh, Australia  
Georgios I. Giannopoulos, Greece  
Agathoklis Giaralis, UK  
Pablo Gil, Spain  
Anna M. Gil-Lafuente, Spain  
Ivan Giorgio, Italy  
Gaetano Giunta, Luxembourg  
Alessio Gizzi, Italy  
Jefferson L.M.A. Gomes, UK  
Emilio Gómez-Déniz, Spain  
Antonio M. Gonçalves de Lima, Brazil  
David González, Spain  
Chris Goodrich, USA  
Rama S. R. Gorla, USA  
Nicolas Gourdain, France  
Kannan Govindan, Denmark  
Antoine Grall, France  
George A. Gravvanis, Greece  
Fabrizio Greco, Italy  
Simonetta Grilli, Italy  
Jason Gu, Canada  
Federico Guarracino, Italy  
Michele Guida, Italy  
José L. Guzmán, Spain  
Quang Phuc Ha, Australia  
Petr Hájek, Czech Republic  
Zhen-Lai Han, China  
Thomas Hanne, Switzerland  
Mohammad A. Hariri-Ardebili, USA  
Xiao-Qiao He, China  
Sebastian Heidenreich, Germany  
Luca Heltai, Italy  
Alfredo G. Hernández-Díaz, Spain  
M.I. Herreros, Spain  
Eckhard Hitzer, Japan  
Paul Honeine, France  
Jaromir Horacek, Czech Republic  
Muneo Hori, Japan  
András Horváth, Italy  
S. Hassan Hosseinnia, Netherlands  
Mengqi Hu, USA  
Gordon Huang, Canada

Sajid Hussain, Canada  
 Asier Ibeas, Spain  
 Orest V. Iftime, Netherlands  
 Przemyslaw Ignaciuk, Poland  
 Giacomo Innocenti, Italy  
 Emilio Insfran Pelozo, Spain  
 Alessio Ishizaka, UK  
 Nazrul Islam, USA  
 Benoit Jung, France  
 Benjamin Ivorra, Spain  
 Payman Jalali, Finland  
 Mahdi Jalili, Australia  
 Łukasz Jankowski, Poland  
 Samuel N. Jator, USA  
 Juan C. Jauregui-Correa, Mexico  
 Reza Jazar, Australia  
 Khalide Jbilou, France  
 Piotr Jędrzejowicz, Poland  
 Isabel S. Jesus, Portugal  
 Linni Jian, China  
 Bin Jiang, China  
 Zhongping Jiang, USA  
 Emilio Jiménez Macías, Spain  
 Ningde Jin, China  
 Xiaoliang Jin, USA  
 Liang Jing, Canada  
 Dylan F. Jones, UK  
 Palle E. Jorgensen, USA  
 Vyacheslav Kalashnikov, Mexico  
 Tamas Kalmar-Nagy, Hungary  
 Tomasz Kapitaniak, Poland  
 Julius Kaplunov, UK  
 Haranath Kar, India  
 Konstantinos Karamanos, Belgium  
 Krzysztof Kecik, Poland  
 Jean-Pierre Kenne, Canada  
 Chaudry M. Khalique, South Africa  
 Do Wan Kim, Republic of Korea  
 Nam-Il Kim, Republic of Korea  
 Jan Koci, Czech Republic  
 Ioannis Kostavelis, Greece  
 Sotiris B. Kotsiantis, Greece  
 Manfred Krafczyk, Germany  
 Frederic Kratz, France  
 Petr Krysl, USA  
 Krzysztof S. Kulpa, Poland  
 Shailesh I. Kundalwal, India  
 Jurgen Kurths, Germany  
 Kyandoghere Kyamakya, Austria  
 Davide La Torre, Italy  
 Risto Lahdelma, Finland  
 Hak-Keung Lam, UK  
 Giovanni Lancioni, Italy  
 Jimmy Lauber, France  
 Antonino Laudani, Italy  
 Hervé Laurent, France  
 Aimé Lay-Ekuakille, Italy  
 Nicolas J. Leconte, France  
 Dimitri Lefebvre, France  
 Eric Lefevre, France  
 Marek Lefik, Poland  
 Yaguo Lei, China  
 Kauko Leiviskä, Finland  
 Thibault Lemaire, France  
 Roman Lewandowski, Poland  
 Chen-Feng Li, China  
 Jian Li, China  
 En-Qiang Lin, USA  
 Zhiyun Lin, China  
 Peide Liu, China  
 Peter Liu, Taiwan  
 Wanquan Liu, Australia  
 Bonifacio Llamazares, Spain  
 Alessandro Lo Schiavo, Italy  
 Jean Jacques Loiseau, France  
 Francesco Lolli, Italy  
 Paolo Lonetti, Italy  
 Sandro Longo, Italy  
 Sebastian López, Spain  
 Pablo Lopez-Crespo, Spain  
 Luis M. López-Ochoa, Spain  
 Ezequiel López-Rubio, Spain  
 Vassilios C. Loukopoulos, Greece  
 Jose A. Lozano-Galant, Spain  
 Helen Lu, Australia  
 Gabriel Luque, Spain  
 Valentin Lychagin, Norway  
 Antonio Madeo, Italy  
 José María Maestre, Spain  
 Alessandro Magnani, Italy  
 Fazal M. Mahomed, South Africa  
 Noureddine Manamanni, France  
 Paolo Manfredi, Italy  
 Didier Maquin, France  
 Giuseppe Carlo Marano, Italy  
 Damijan Markovic, France  
 Francesco Marotti de Sciarra, Italy  
 Rui Cunha Marques, Portugal  
 Luis Martínez, Spain  
 Rodrigo Martinez-Bejar, Spain  
 Guiomar Martín-Herrán, Spain  
 Denizar Cruz Martins, Brazil  
 Benoit Marx, France  
 Elio Masciari, Italy  
 Franck Massa, France  
 Paolo Massioni, France  
 Alessandro Mauro, Italy  
 Fabio Mazza, Italy  
 Laura Mazzola, Italy  
 Driss Mehdi, France  
 Roderick Melnik, Canada  
 Pasquale Memmolo, Italy  
 Xiangyu Meng, USA  
 Jose Merodio, Spain  
 Alessio Merola, Italy  
 Mahmoud Mesbah, Iran  
 Luciano Mescia, Italy  
 Laurent Mevel, France  
 Mariusz Michta, Poland  
 Leandro F. F. MIGUEL, Brazil  
 Aki Mikkola, Finland  
 Giovanni Minafò, Italy  
 Hiroyuki Mino, Japan  
 Pablo Mira, Spain  
 D. Mitsotakis, New Zealand  
 Vito Mocella, Italy  
 Sara Montagna, Italy  
 Roberto Montanini, Italy  
 Francisco J. Montáns, Spain  
 Luiz H. A. Monteiro, Brazil  
 Gisele Mophou, France  
 Rafael Morales, Spain  
 Marco Morandini, Italy  
 Javier Moreno-Valenzuela, Mexico  
 Simone Morganti, Italy  
 Caroline Mota, Brazil  
 Aziz Moukrim, France  
 Dimitris Mourtzis, Greece  
 Emiliano Mucchi, Italy  
 Josefa Mula, Spain  
 Jose J. Muñoz, Spain

Giuseppe Muscolino, Italy  
Marco Mussetta, Italy  
Hakim Naceur, France  
Alessandro Naddeo, Italy  
Hassane Naji, France  
Mariko Nakano-Miyatake, Mexico  
Keivan Navaie, UK  
AMA Neves, Portugal  
Luís C. Neves, UK  
Dong Ngoduy, New Zealand  
Nhon Nguyen-Thanh, Singapore  
Tatsushi Nishi, Japan  
Xesús Nogueira, Spain  
Ben T. Nohara, Japan  
Mohammed Nouari, France  
Mustapha Nourelfath, Canada  
Włodzimierz Ogryczak, Poland  
Roger Ohayon, France  
Krzysztof Okarma, Poland  
Mitsuhiro Okayasu, Japan  
Alberto Olivares, Spain  
Enrique Onieva, Spain  
Calogero Orlando, Italy  
Alejandro Ortega-Moñux, Spain  
Sergio Ortobelli, Italy  
Naohisa Otsuka, Japan  
Erika Ottaviano, Italy  
Pawel Packo, Poland  
Arturo Pagano, Italy  
Alkis S. Paipetis, Greece  
Roberto Palma, Spain  
Alessandro Palmeri, UK  
Pasquale Palumbo, Italy  
Jürgen Pannek, Germany  
Elena Panteley, France  
Achille Paolone, Italy  
George A. Papakostas, Greece  
Xosé M. Pardo, Spain  
Vicente Parra-Vega, Mexico  
Manuel Pastor, Spain  
Petr Páta, Czech Republic  
Pubudu N. Pathirana, Australia  
Surajit K. Paul, India  
Sitek Paweł, Poland  
Luis Payá, Spain  
Alexander Paz, USA  
Igor Pažanin, Croatia  
Libor Pekař, Czech Republic  
Francesco Pellicano, Italy  
Marcello Pellicciari, Italy  
Haipeng Peng, China  
Mingshu Peng, China  
Zhengbiao Peng, Australia  
Zhi-ke Peng, China  
Marzio Pennisi, Italy  
Maria Patrizia Pera, Italy  
Matjaz Perc, Slovenia  
A. M. Bastos Pereira, Portugal  
Ricardo Perera, Spain  
Francesco Pesavento, Italy  
Ivo Petras, Slovakia  
Francesco Petrini, Italy  
Lukasz Pieczonka, Poland  
Dario Piga, Switzerland  
Paulo M. Pimenta, Brazil  
Antonina Pirrotta, Italy  
Marco Pizzarelli, Italy  
Vicent Pla, Spain  
Javier Plaza, Spain  
Dragan Poljak, Croatia  
Jorge Pomares, Spain  
Sébastien Poncet, Canada  
Volodymyr Ponomaryov, Mexico  
Jean-Christophe Ponsart, France  
Mauro Pontani, Italy  
Cornelio Posadas-Castillo, Mexico  
Francesc Pozo, Spain  
Christopher Pretty, New Zealand  
Luca Pugi, Italy  
Krzysztof Puszynski, Poland  
Giuseppe Quaranta, Italy  
Vitomir Racic, Italy  
Jose Ragot, France  
Carlo Rainieri, Italy  
K. Ramamani Rajagopal, USA  
Ali Ramazani, USA  
Higinio Ramos, Spain  
Alain Rassineux, France  
S.S. Ravindran, USA  
Alessandro Reali, Italy  
Jose A. Reinoso, Spain  
Oscar Reinoso, Spain  
Fabrizio Renno, Italy  
Nidhal Rezg, France  
Ricardo Riaza, Spain  
Francesco Riganti-Fulginei, Italy  
Gerasimos Rigatos, Greece  
Francesco Ripamonti, Italy  
Jorge Rivera, Mexico  
Eugenio Roanes-Lozano, Spain  
Bruno G. M. Robert, France  
Ana Maria A. C. Rocha, Portugal  
José Rodellar, Spain  
Luigi Rodino, Italy  
Rosana Rodríguez López, Spain  
Ignacio Rojas, Spain  
Alessandra Romolo, Italy  
Debasish Roy, India  
Gianluigi Rozza, Italy  
Rubén Ruiz, Spain  
Antonio Ruiz-Cortes, Spain  
Ivan D. Rukhlenko, Australia  
Mazen Saad, France  
Kishin Sadarangani, Spain  
Andrés Sáez, Spain  
Mehrddad Saif, Canada  
John S. Sakellariou, Greece  
Salvatore Salamone, USA  
Vicente Salas, Spain  
Jose V. Salcedo, Spain  
Nunzio Salerno, Italy  
Miguel A. Salido, Spain  
Roque J. Saltarén, Spain  
Alessandro Salvini, Italy  
Sylwester Samborski, Poland  
Ramon Sancibrian, Spain  
Giuseppe Sanfilippo, Italy  
Miguel A. F. Sanjuan, Spain  
Vittorio Sansalone, France  
José A. Sanz-Herrera, Spain  
Nickolas S. Sapidis, Greece  
Evangelos J. Sapountzakis, Greece  
Luis Saucedo-Mora, Spain  
Marcelo A. Savi, Brazil  
Andrey V. Savkin, Australia  
Roberta Sburlati, Italy  
Gustavo Scaglia, Argentina  
Thomas Schuster, Germany  
Oliver Schütze, Mexico  
Lotfi Senhadji, France  
Junwon Seo, USA

Joan Serra-Sagrsta, Spain  
Gerardo Severino, Italy  
Ruben Sevilla, UK  
Stefano Sfarra, Italy  
Mohamed Shaat, Egypt  
Mostafa S. Shadloo, France  
Leonid Shaikhet, Israel  
Hassan M. Shanechi, USA  
Bo Shen, Germany  
Suzanne M. Shontz, USA  
Babak Shotorban, USA  
Zhan Shu, UK  
Nuno Simões, Portugal  
Christos H. Skiadas, Greece  
Konstantina Skouri, Greece  
Neale R. Smith, Mexico  
Bogdan Smolka, Poland  
Delfim Soares Jr., Brazil  
Alba Sofi, Italy  
Giovanni Solari, Italy  
Francesco Soldovieri, Italy  
Raffaele Solimene, Italy  
Jussi Sopanen, Finland  
Marco Spadini, Italy  
Bernardo Spagnolo, Italy  
Paolo Spagnolo, Italy  
Ruben Specogna, Italy  
Vasilios Spitas, Greece  
Sri Sridharan, USA  
Ivanka Stamova, USA  
Rafał Stanisławski, Poland  
Florin Stoican, Romania  
Salvatore Strano, Italy  
Yakov Strelniker, Israel  
Guang-Yong Sun, China  
Sergey A. Suslov, Australia  
Thomas Svensson, Sweden  
Andrzej Swierniak, Poland  
Andras Szekrenyes, Hungary  
Kumar K. Tamma, USA  
Yang Tang, Germany  
Hafez Tari, USA  
Alessandro Tasora, Italy  
Sergio Teggi, Italy  
Ana C. Teodoro, Portugal

Tai Thai, Australia  
Alexander Timokha, Norway  
Gisella Tomasini, Italy  
Francesco Tornabene, Italy  
Antonio Tornambe, Italy  
Javier Martinez Torres, Spain  
Mariano Torrisi, Italy  
George Tsiatas, Greece  
Antonios Tsourdos, UK  
Federica Tubino, Italy  
Nerio Tullini, Italy  
Andrea Tundis, Italy  
Emilio Turco, Italy  
Vladimir Turetsky, Israel  
Mustafa Tutar, Spain  
Ilhan Tuzcu, USA  
Efstratios Tzirtzilakis, Greece  
Filippo Ubertini, Italy  
Francesco Ubertini, Italy  
Mohammad Uddin, Australia  
Hassan Ugail, UK  
Giuseppe Vairo, Italy  
Eusebio Valero, Spain  
Pandian Vasant, Malaysia  
Marcello Vasta, Italy  
Carlos-Renato Vázquez, Mexico  
Miguel E. Vázquez-Méndez, Spain  
Josep Vehi, Spain  
Martin Velasco Villa, Mexico  
K. C. Veluvolu, Republic of Korea  
Fons J. Verbeek, Netherlands  
Franck J. Vernerey, USA  
Georgios Veronis, USA  
Vincenzo Vespri, Italy  
Renato Vidoni, Italy  
V. Vijayaraghavan, Australia  
Anna Vila, Spain  
Rafael J. Villanueva, Spain  
Francisco R. Villatoro, Spain  
Uchechukwu E. Vincent, UK  
Gareth A. Vio, Australia  
Francesca Vipiana, Italy  
Stanislav Vítek, Czech Republic  
Thuc P. Vo, UK  
Jan Vorel, Czech Republic

Michael Vynnycky, Sweden  
Hao Wang, USA  
Liliang Wang, UK  
Shuming Wang, China  
Yongqi Wang, Germany  
Roman Wan-Wendner, Austria  
Jaroslaw Wąs, Poland  
P.H. Wen, UK  
Waldemar T. Wójcik, Poland  
Changzhi Wu, Australia  
Desheng D. Wu, Sweden  
Hong-Yu Wu, USA  
Yuqiang Wu, China  
Michalis Xenos, Greece  
Guangming Xie, China  
Xue-Jun Xie, China  
Gen Q. Xu, China  
Hang Xu, China  
Joseph J. Yame, France  
Xinggong Yan, UK  
Mijia Yang, USA  
Yongheng Yang, Denmark  
Luis J. Yebra, Spain  
Peng-Yeng Yin, Taiwan  
Qin Yuming, China  
Elena Zaitseva, Slovakia  
Arkadiusz Żak, Poland  
Daniel Zaldivar, Mexico  
Francesco Zammori, Italy  
Vittorio Zampoli, Italy  
Rafał Zdunek, Poland  
Ibrahim Zeid, USA  
Huaguang Zhang, China  
Kai Zhang, China  
Qingling Zhang, China  
Xian-Ming Zhang, Australia  
Xuping Zhang, Denmark  
Zhao Zhang, China  
Yifan Zhao, UK  
Jian G. Zhou, UK  
Quanxin Zhu, China  
Mustapha Zidi, France  
Gaetano Zizzo, Italy

# Contents

---

## **Advances in Variational and Partial Differential Equation-Based Models for Image Processing and Computer Vision**

Tudor Barbu , Gabriela Marinoschi, Costică Moroşanu, and Ionuţ Munteanu  
Editorial (2 pages), Article ID 1701052, Volume 2018 (2018)

## **A Lightweight Surface Reconstruction Method for Online 3D Scanning Point Cloud Data Oriented toward 3D Printing**

Buyun Sheng , Feiyu Zhao , Xiyan Yin, Chenglei Zhang, Hui Wang, and Peide Huang  
Research Article (16 pages), Article ID 4673849, Volume 2018 (2018)

## **An Improved Fractional-Order Optical Flow Model for Motion Estimation**

Bin Zhu , Lianfang Tian, Qiliang Du , Qiuxia Wu , and Lixin Shi   
Research Article (6 pages), Article ID 6278719, Volume 2018 (2018)

## **Fast Video Dehazing Using Per-Pixel Minimum Adjustment**

Zhong Luan , Hao Zeng, Yuanyuan Shang , Zhuhong Shao, and Hui Ding  
Research Article (8 pages), Article ID 9241629, Volume 2018 (2018)

## **Efficient 3D Volume Reconstruction from a Point Cloud Using a Phase-Field Method**

Darae Jeong , Yibao Li, Heon Ju Lee, Sang Min Lee, Junxiang Yang, Seungwoo Park, Hyundong Kim, Yongho Choi , and Junseok Kim   
Research Article (9 pages), Article ID 7090186, Volume 2018 (2018)

## **Image Regularity and Fidelity Measure with a Two-Modality Potential Function**

Weiwei Wang, Shengjiang Kong, Amir Razi, and Xiangchu Feng  
Research Article (14 pages), Article ID 1408910, Volume 2017 (2018)

## Editorial

# Advances in Variational and Partial Differential Equation-Based Models for Image Processing and Computer Vision

**Tudor Barbu** <sup>1</sup>, **Gabriela Marinoschi**,<sup>2</sup> **Costică Moroşanu**,<sup>3</sup> and **Ionuţ Munteanu**<sup>3</sup>

<sup>1</sup>*Institute of Computer Science of the Romanian Academy, Iaşi Branch, Romania*

<sup>2</sup>*Institute of Mathematical Statistics and Applied Mathematics of the Romanian Academy, Romania*

<sup>3</sup>*Faculty of Mathematics, “Al. I. Cuza” University, Iaşi, Romania*

Correspondence should be addressed to Tudor Barbu; [tudor.barbu@iit.academiaromana-is.ro](mailto:tudor.barbu@iit.academiaromana-is.ro)

Received 12 March 2018; Accepted 13 March 2018; Published 5 June 2018

Copyright © 2018 Tudor Barbu 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.

Partial differential equation- (PDE-) based models have led to an entire new subdomain of image processing and analysis. The partial differential equations express continuous change, so they have long been used to formulate dynamical phenomena in many important engineering domains. Thus, they have proved their usefulness in various image processing and computer vision fields, where they have been widely applied in the last 30 years, since they offer some important advantages to these areas, such as their modelling flexibility, their strong mathematical foundation, and their numerical approximations representing reinterpretations of numerous classical image processing techniques.

Since it is very common in image processing and analysis to obtain PDE models from variational problems involving functional minimization, many PDE variational techniques have been developed in these domains. They have important advantages in both theory and computation, compared with other techniques. Variational models can achieve high speed, accuracy, and also stability. While many PDE-based image processing models follow variational principles, there also exist such PDE schemes that are not derived from variational approaches. Both variational and nonvariational partial differential equations appear in a variety of image processing and computer vision areas, successfully answering the challenges that still persist in these domains.

Thus, the nonlinear second- and fourth-order diffusion-based models represent the best denoising and restoration solution, since they remove successfully various types of noise while overcoming the undesirable effects and preserving the image details, which still constitutes a challenge in this area.

Variational and nonlinear second-, third-, and fourth-order PDE-based techniques are also applied successfully in the structural inpainting domain and the image compression area that uses the inpainting results in the decompression stage. Also, a lot of effective PDE variational algorithms have been developed in the image segmentation and registration domains. Since the optical flow is successfully computed using variational methods, the video motion estimation and computer vision fields like object detection and tracking represent also important application areas of partial differential equations.

The main purpose of this special issue is to gather scientific works disseminating advanced research on several topics related to these PDE-based image processing and computer vision areas. We have received a total of 20 submissions, from authors in many countries all around the world. Since only a few of these papers have been considered appropriate enough to be published in the journal, by our editorial team, this special issue is composed of five peer-reviewed original research articles.

In the paper “Fast Video Dehazing Using Per-Pixel Minimum Adjustment”, Z. Luan et al. introduce a fast video dehazing method which represents an image restoration problem. A computer vision scheme that is atmospheric scattering model is used for haze removal. The atmospheric light from this model is estimated by using a quad-tree based method. The proposed technique improves the efficiency of video dehazing and outperforms other haze removal methods.

In the paper “Efficient 3D Volume Reconstruction from a Point Cloud Using a Phase-Field Method”, D. Jeong et al.

propose a PDE-based efficient 3D volume reconstruction technique from unorganized point clouds, using a phase-field method. The reconstruction is based on the 3D binary image segmentation method using a modified Allen-Cahn equation. The effectiveness of the proposed algorithm has been proved by the successful computational experiments.

The paper “Image Regularity and Fidelity Measure with a Two-Modality Potential Function”, authored by W. Wang et al., describes a PDE variational model for image restoration. The authors define a strictly convex smooth potential function for this model and use it to measure the data fidelity and the regularity for image denoising and cartoon texture decomposition. Given the mathematical properties of this potential function, the proposed variational restoration approach outperforms the total variation based denoising models and works successfully for many categories of noise including Gaussian noise, impulse noise, Poisson noise, and mixed noise.

In the paper “A Lightweight Surface Reconstruction Method for Online 3D Scanning Point Cloud Data Oriented toward 3D Printing”, B. Sheng et al. provide an online lightweight surface reconstruction approach composed of three algorithms: a point cloud update algorithm (PCU), a rapid iterative closest point algorithm (RICP), and an improved Poisson surface reconstruction algorithm (IPSR) that uses biharmonic-like fourth-order PDEs to repair the mesh holes on the reconstructed lightweight mesh. An online personalized customization system oriented toward 3D printing is then constructed using the proposed approach.

In the paper entitled “An Improved Fractional-order Optical Flow Model for Motion Estimation”, B. Zhu et al. propose a fractional-order optical flow model that improves the Horn and Schunck optical flow model. The considered variational model for video motion estimation substitutes the brightness constraint equation of the HS model with the fractional-order Taylor series expansion and gets a fractional-order brightness constraint equation. The performed experiments demonstrate the performance of this improved variational optical flow model.

## Acknowledgments

We would like to thank all the authors who kindly contributed their research papers to this special issue. We also thank the reviewers for their assistance during the preparation and publication of our special issue.

*Tudor Barbu  
Gabriela Marinoschi  
Costică Moroşanu  
Ionuţ Munteanu*

## Research Article

# A Lightweight Surface Reconstruction Method for Online 3D Scanning Point Cloud Data Oriented toward 3D Printing

Buyun Sheng <sup>1,2</sup>, Feiyu Zhao <sup>1</sup>, Xiyan Yin,<sup>1</sup> Chenglei Zhang,<sup>1</sup>  
Hui Wang,<sup>1</sup> and Peide Huang<sup>1</sup>

<sup>1</sup>School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan, Hubei 430070, China

<sup>2</sup>Hubei Digital Manufacturing Key Laboratory, Wuhan University of Technology, Wuhan, Hubei 430070, China

Correspondence should be addressed to Buyun Sheng; shengby@whut.edu.cn

Received 16 November 2017; Revised 23 February 2018; Accepted 4 March 2018; Published 16 May 2018

Academic Editor: Tudor Barbu

Copyright © 2018 Buyun Sheng 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 existing surface reconstruction algorithms currently reconstruct large amounts of mesh data. Consequently, many of these algorithms cannot meet the efficiency requirements of real-time data transmission in a web environment. This paper proposes a lightweight surface reconstruction method for online 3D scanned point cloud data oriented toward 3D printing. The proposed online lightweight surface reconstruction algorithm is composed of a point cloud update algorithm (PCU), a rapid iterative closest point algorithm (RICP), and an improved Poisson surface reconstruction algorithm (IPSR). The generated lightweight point cloud data are pretreated using an updating and rapid registration method. The Poisson surface reconstruction is also accomplished by a pretreatment to recompute the point cloud normal vectors; this approach is based on a least squares method, and the postprocessing of the PDE patch generation was based on biharmonic-like fourth-order PDEs, which effectively reduces the amount of reconstructed mesh data and improves the efficiency of the algorithm. This method was verified using an online personalized customization system that was developed with WebGL and oriented toward 3D printing. The experimental results indicate that this method can generate a lightweight 3D scanning mesh rapidly and efficiently in a web environment.

## 1. Introduction

3D printing is a technology that manufacture solid parts through accumulating material layer by layer [1], and it can efficiently support the recent accomplishments in 3D digital-to-solid modeling [2]. 3D printing can produce customized objects [3, 4] that can be realized by any merchant to personalize products for a consumer, meet quality expectations, and add personal refinements [5]. With the development of the Internet, 3D printing cloud services platform can provide integrated 3D printing services [6] to serve customers at all levels: from cloud-based 3D scanning of a physical model to personalized and/or customized products [7]. However, cloud-based point cloud data surface reconstruction from 3D scanned point clouds requires not only a low-complexity reconstruction algorithm to meet the high concurrency requirements of cloud service platforms but also the reconstruction of a lightweight mesh to meet the

requirements of efficient and real-time data transmission in web environments [8]. Most of the existing surface reconstruction algorithms use an accurate mesh as a reconstruction target without considering a reduction in the complexity of reconstruction algorithms or the lightweight nature of the 3D model data. In addition, a PC connected to a 3D scanner cannot receive the generated point cloud data directly from the cloud server due to Internet security requirements. Thus, real-time point cloud data transmission from the 3D scanner to the cloud service platform cannot be realized along with the process of 3D scanning [9].

To reduce the complexity of the algorithm and generate a lightweight 3D model, in this paper, an online lightweight surface reconstruction algorithm is proposed, which is composed of a point cloud update algorithm (PCU), a rapid iterative closest point algorithm (RICP), and an improved Poisson surface reconstruction algorithm (IPSR). The PCU is used to obtain the latest point cloud data generated by the 3D

scanner in real time. It can remove part of the noise from the point cloud data and makes it lightweight using a filtration approach based on the center of gravity. The 3D scanner used in this research is called Ciclop, which independently generates two point cloud data sets through its two infrared laser modules. Rapid and accurate registration between the two sets of point cloud data is achieved using the RICP. The resulting modified point cloud set restores the outline of the scanned object more accurately. During the process of surface reconstruction by the IPSR, the normal vectors of the point cloud data are pretreated via a recomputation method based on the least squares method. To repair the mesh holes that can easily be generated by the Poisson surface reconstruction algorithm, an iterative postprocessing algorithm for PDE patch generation based on biharmonic-like fourth-order PDEs is executed successively in IPSR, which also reduces the amount of reconstructed mesh data.

To realize real-time point cloud data transmission, a dynamic visualization framework for point cloud data, based on WebSocket, is also proposed in this paper. The point cloud data generated by Ciclop in real-time can be encapsulated as a JSON file in the local server and dynamically displayed in the browser using WebSocket and WebGL. In addition, the fluency and high rendering effect of dynamic visualization of point cloud data is ensured by the Web Worker mechanism in high concurrency environments.

In sum, the novel contributions in this paper are as follows: (1) an online lightweight surface reconstruction algorithm, in which the lightweight operations are conducted at every step, from point cloud data acquisition to preprocessing and to surface reconstruction. This reconstruction ensures that a reduced data volume of the 3D model meets the requirement for web-based data transmission; (2) a dynamic visualization framework for point cloud data based on WebSocket, which achieves online dynamic visualization of point cloud data in high concurrency environments; (3) an online personalized customization system oriented toward 3D printing, which dynamically visualizes the point cloud data through the 3D scanning process and the efficient and rapid reconstruction of the lightweight mesh in the web environment.

The remainder of this paper is organized as follows. Section 2 reviews prior relevant research concerning surface reconstruction. Section 3 elaborates the online lightweight surface reconstruction algorithm, which includes the PCU, the RICP, and the IPSR. Section 4 presents the experiments, which demonstrate the system framework and experimental platform, point cloud update and the results of dynamic visualization tests, point cloud registration tests, and online surface reconstruction tests, and Section 5 concludes the paper.

## 2. Related Works

*2.1. Surface Reconstruction.* In the field of surface reconstruction, Zhang et al. [10] proposed a new approach to simultaneously denoise and parameterize unorganized point cloud data, in which the key ingredient was an “as-rigid-as-possible” meshless parameterization that maps a point cloud

with disk topology to a 3D plane. Denoising and reconstruction of the point cloud are executed in the same process. Huang et al. [11] proposed a 3D reconstruction system that performs fast 3D modeling using a Kinect sensor and can automatically detect the face region and track head pose using an ICP algorithm. A volumetric integration method was used to fuse the resulting data. Finally, a marching cubes algorithm was used to reconstruct the face model used for display, which effectively improved the face model accuracy. Centin et al. [12] proposed a new Poisson reconstruction algorithm based on an interpolation method and exploited it to efficiently guide a restricted Delaunay framework protecting the input mesh and the boundary curves. The boundary of a repaired hole can be seamlessly integrated with the patch boundary. Garrett et al. [13] proposed an accurate object-tracking method based on an ICP algorithm. Reduced-density friendly point clouds can be reconstructed to create an accurate mesh through a Poisson surface reconstruction algorithm via object tracking. Peyrot et al. [14] proposed a framework to design semiregular meshes directly from stereoscopic images in which feature-preserving samples of the stereoscopic images were extracted to obtain a base mesh. Then, using iterative procedures, a semiregular mesh of the original surface was generated from the base mesh. Liu et al. [15] proposed a method for representing 3D outdoor scenes via 3D laser point clouds that used a fast optimal bearing angle (FOBA) approach to project the 3D laser point clouds to 2D images, which greatly reduced the computational cost of scene segmentation with little loss of accuracy, thus improving the efficiency of the reconstruction. Li et al. [16] proposed a Prominent Cross-Section algorithm embedded with a curvature constraint that can automatically identify the boundary of a damaged area, thereby eliminating any defective point clouds during the reconstruction process. They also proposed an improved iterative ICP algorithm to automatically identify and eliminate any unreliable corresponding pairs. Roth et al. [17] proposed a method for reconstructing the 3D surface model of an individual’s face along with albedo information. A 3D Morphable Model was fitted to form a personalized template, and a novel photometric stereo formulation was also developed. This formulation of an accurate 3D face model can be generated from low-quality photo collections and with fewer images based on the albedo information. Boltcheva and Lévy [18] proposed a method for reconstructing a 3D surface triangulation from an input point set based on a restricted Voronoi diagram. The properties of the restricted Voronoi cells were utilized to make an embarrassingly parallel implementation that could process 100 million vertices within a few minutes to achieve rapid mesh reconstruction.

*2.2. Network Communications.* In the field of network communication, Marion and Jomier [19] proposed a web-based system that focused on collaborative interaction. The system is composed of two innovative technologies: WebGL and WebSocket. The architecture of the proposed system was presented, and the operating process of the system was further elaborated in the field of telemedicine. Zimmer and Kerren [20] suggested a client/server-based visualization

system for collaboratively exploring graphs in which the web application was rendered by WebGL, and the real-time visualization transmission was achieved by WebSocket. Mwalongo et al. [21] proposed an approach to visualize dynamic molecular data using WebGL. The approach exploits HTML5 technologies such as WebSocket and Web Workers and used efficient data encoding techniques to minimize the data transferred to the client. This approach allows scientists to perform analyses from dynamic visualizations of the molecular structure data via a browser. Renambot et al. [22] presented SAGE2, a software framework based on WebSocket that enables local and remote collaboration. 3D-rendering and cross-platform visualization and interaction were supported in this framework, which can be utilized in education, academic conferences, and so on.

**2.3. Existing Problems in Current Research.** Although any of the current surface reconstruction algorithms [11–14] can reconstruct accurate meshes, the data volume of 3D meshes is larger; thus, the resulting average reconstruction time is also longer. As an example of this increased data volume, for Stanford Bunny point cloud data (1.89 MB), the reconstruction mesh was at least 20 MB and the average reconstruction time was at least 15 s. Although the quality of the point cloud and mesh improved while the reconstruction time decreased due to denoising and registration pretreatments, the surface reconstruction algorithms [10, 16] were not optimized, resulting in an insufficiently lightweight mesh. Consequently, the reconstruction was unable to meet the requirements for network transmission. While the methods of real-time data communication technologies, based on WebSocket [19–22], were achieved only if the effects of the relevant data were stored in a database, the data could be visualized dynamically in the browser. Thus, the real-time data communication between device and cloud server was not achieved. Overall, dynamic visualization of point cloud data in a browser, processed by 3D scanning technology, is difficult to achieve.

### 3. Online Lightweight Surface Reconstruction Algorithm

The method proposed in this paper is achieved by using an open-source 3D scanner called Ciclop. The structure of Ciclop is shown in Figure 1. The two point cloud sets, denoted by  $P_A = \{p_{A_1}, p_{A_2}, \dots, p_{A_n}\}$  and  $P_B = \{p_{B_1}, p_{B_2}, \dots, p_{B_n}\}$ , are generated from infrared laser modules A and B, respectively.

To ensure the efficiency and speed of surface reconstruction and the reconstruction of a lightweight mesh, an online lightweight surface reconstruction algorithm is proposed in this paper. The algorithm includes the PCU, the RICP, and the IPSR. These three subalgorithms are executed in turn. A flow-chart of the online lightweight surface reconstruction algorithm is shown in Figure 2.

The process in the PCU can be described as two point cloud sets,  $P_A$  and  $P_B$ , which are generated by the infrared laser module A and B, respectively, from a set of depth images  $I = \{I_1, I_2, \dots, I_i, \dots, I_n\}$  which are acquired continuously

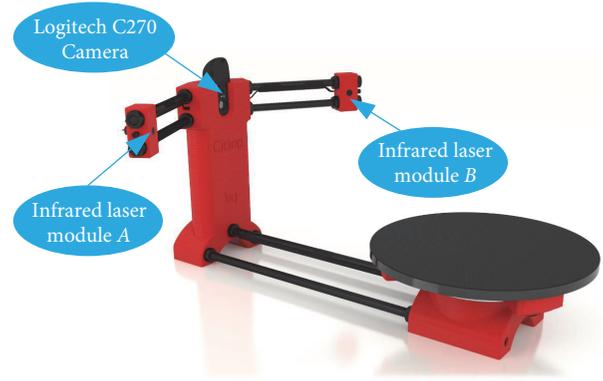


FIGURE 1: Open source Ciclop 3D scanner.

from Ciclop's Logitech C270 camera. The point cloud data is updated during the process of 3D scanning. The determination of whether to terminate the process is executed after each iteration. The holonomic point cloud sets  $P_A$  and  $P_B$  should be generated over several iterations.

The RICP performs the registration between  $P_A$  and  $P_B$ . First,  $P_A$  is selected as the benchmark, and a preregistration of  $P_A$  and  $P_B$  is executed based on the spatial relationship between the infrared laser modules A and B. Thus,  $P_B$  is rotated to the position of  $[P_B]$ . Then, the ICP algorithm transforms point cloud set  $[P_B]$ , rotating it to a new position,  $\overline{P}_B$ . Finally, the solution of modified point cloud set  $P^*$  is conducted using the spatial relationship between  $P_A$  and  $\overline{P}_B$ .

The IPSR first executes a pretreatment step that reconstructs the normal vectors of the modified point cloud set  $P^*$  based on the least squares method. Subsequently, the Poisson surface reconstruction and a postprocessing step that generates PDE patches based on a biharmonic-like fourth-order PDEs are conducted successively. The lightweight mesh  $M$  is generated through the above procedures.

**3.1. Point Cloud Update Algorithm.** A point cloud update algorithm (PCU) is proposed in this paper. The updating process involves the following process: whenever the Ciclop turntable is rotated by 1.8 degrees, a new depth image  $I_i$  is generated by the Logitech C270, adding the new feature points  $p_i$  and  $p_j$  to the point cloud sets  $P_{A_i}$  and  $P_{B_i}$ , respectively.  $P_A$  and  $P_B$  will be refined after several iterations.

To meet the requirements of efficient and real-time data transmission based on B/S, the new feature points generated from  $I_i$  should be filtered before they are added into the existing point cloud sets. As shown in Figure 3, first, new feature points are selected. Then, new triangular structures are formed based on these new feature points and the base of a triangular structure. These new triangular structures are considered the base triangular structures that need further refinement.

The filtration approach is conducted based on a center of gravity approach that can be described as follows.

*Step 1.* Three feature points  $p_A, p_B, p_C$  that are about to be reconstructed as mesh triangle are obtained from the depth

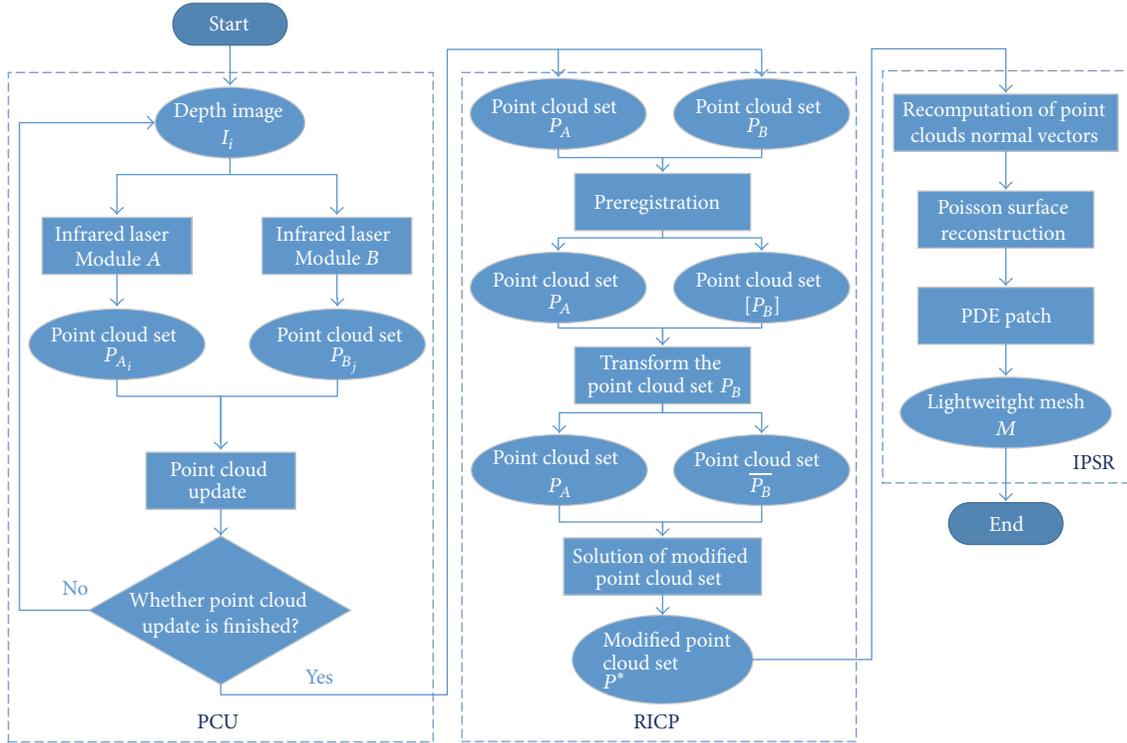


FIGURE 2: Online lightweight surface reconstruction algorithm.

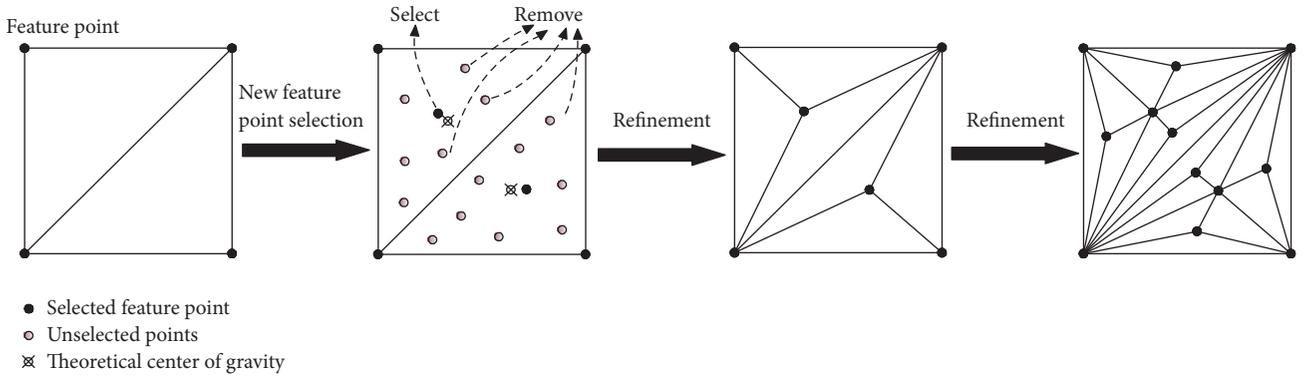


FIGURE 3: Filtration of new feature points.

image  $I_i$ , and their 2D coordinates  $p_A(x_A, y_A)$ ,  $p_B(x_B, y_B)$ , and  $p_C(x_C, y_C)$  are extracted. Their center of gravity is computed by  $p_G = (x_G, y_G) = ((x_A + x_B + x_C)/3, (y_A + y_B + y_C)/3)$ .

*Step 2.* The 2D points obtained from the depth image  $I_{i+1}$  are imported into the existing point cloud set. Any points located inside the triangle (ignoring the points on the boundary) are selected and collected as the point set  $\{p_1, p_2, \dots, p_n\}$ .

*Step 3.* The variable  $d_{\min}$  is defined as the minimum distance. A bubble sort algorithm [23] is utilized to compute the point ( $p_i$ ) nearest to  $p_G$ , which is determined using the distance formula  $d_{\min} = \sqrt{(x_G - x_i)^2 + (y_G - y_i)^2}$ .

*Step 4.* The new feature point  $p_i$  is selected, and three sub-triangles are generated from this new feature point with the other three existing feature points, after which the filtration process is complete.

Compared with the point cloud update approach that adds all the points into the point cloud set, our approach effectively achieves a denoised and lightweight point cloud to some extent.

*3.2. Rapid Iterative Closest Point Algorithm.* Fundamentally, the RICP can be regarded as a solution of a least squares problem. The method was proposed by Besl and McKay [24], who determined that this approach can accomplish

registration between two point cloud sets  $P_A$  and  $P_B$  [25]. Let  $P_A = \{p_{A_0}, p_{A_1}, \dots, p_{A_n}\}$  be the reference point cloud set and  $P_B = \{p_{B_1}, p_{B_2}, \dots, p_{B_n}\}$  be the sample point cloud set. A translation vector  $\mathbf{t}$  and a rotation matrix  $R$  which transform  $P_B$  to  $P_A$  can be computed by ICP.

To achieve rapid registration between different point cloud sets and reduce the number of iterations, a pre-registration process is conducted between  $P_A$  and  $P_B$ , i.e., the rotation matrix  $R'$  and translation vector  $\mathbf{t}'$  forming a rigid transformation from the position  $O_B(x_B, y_B, z_B)$  to the position  $O_A(x_A, y_A, z_A)$  of the respective infrared laser modules,  $B$  and  $A$ , are computed in a world-coordinate system. Then,  $P_B$  is preassociated with  $R'$  and  $\mathbf{t}'$ , which greatly reduces the time to find the nearest points between two point cloud sets and number of ICP iterations.

(1) *Preregistration.* Translation vector  $\mathbf{t}'$  is computed from  $O_A(x_A, y_A, z_A)$ ,  $O_B(x_B, y_B, z_B)$  and formulated as follows:

$$\mathbf{t}' = O_A - O_B, \quad (1)$$

where  $O_A$  and  $O_B$  should be expressed as pure quaternions (i.e.,  $O'_A(x_A, y_A, z_A, 0)$  and  $O'_B(x_B, y_B, z_B, 0)$ ). Let the origin of the world-coordinate system be  $O$ ; then, the rotation unit quaternion from  $\mathbf{OO}'_B$  to  $\mathbf{OO}'_A$  can be computed as  $q(q_x, q_y, q_z, q_w)$ . According to [26],

$$\mathbf{OO}'_A = q \cdot \mathbf{OO}'_B \cdot q^{-1}, \quad (2)$$

where  $q^{-1}$  is the conjugate quaternion of  $q$  and expressed as  $q^{-1}(-q_x, -q_y, -q_z, q_w)$ . Thus,  $q$  can be solved from (2), and  $q$  can be transformed to the rotation matrix  $R'$ , which is formulated as follows:

$$R' = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix}. \quad (3)$$

Thus, the rigid transformation of  $P_B$  based on  $R'$  and  $\mathbf{t}'$  is implemented to transform  $P_B$  to the point cloud set  $[P_B]$ , which is formulated as follows:

$$[P_B] = R' P_B + \mathbf{t}'. \quad (4)$$

(2) *The Transformation of the Point Cloud Set.* The ICP relies on identifying the nearest neighbors between both point sets. To find the matches between  $P_A$  and  $[P_B]$ , a  $k$ -nearest neighbors search [27] is used that adopts Euclidean distance as the distance measure:

$$\text{dist}([P_B], P_A) = \sqrt{([p_{B_1}] - p_{A_1})^2 + ([p_{B_2}] - p_{A_2})^2 + \dots + ([p_{B_n}] - p_{A_n})^2}. \quad (5)$$

Only one neighbor is returned when  $P_A$  and  $[P_B]$  are in exact spatial correspondence. When too many points are

close (relative to an algorithmic threshold), the match is rejected in the current frame and not considered further. Therefore, a *kd-tree* data structure is used for all points, and an *approximate nearest neighbor search* [28] is implemented to accelerate the search.

All the associated points should be weighted. The weight value is based on the comparability of normal vectors and formulated as follows:

$$w_{ij} = \mathbf{n}_i \cdot \mathbf{n}_j. \quad (6)$$

The normal vectors of two matching points must be aligned in a similar direction after the two point clouds are approximately aligned. The output of this step is a point cloud set consisting of the  $N$  corresponding points in  $P_A$  and  $[P_B]$ , along with the weight values  $w_{ij}$  for each match  $0 \leq w_{ij} \leq 1$ .

To transform the point sets, the translation vector  $\mathbf{t}$  and rotation matrix  $R$  are computed from  $[P_B]$  to  $P_A$ . The translation vector  $\mathbf{t}$  can be computed as the difference between the corresponding centroids of  $P_A$  and  $[P_B]$ , which can be formulated as follows:

$$g_{P_A} = \frac{1}{N_{P_A}} \sum_{i=1}^{N_{P_A}} P_{A_i}$$

$$g_{[P_B]} = \frac{1}{N_{[P_B]}} \sum_{j=1}^{N_{[P_B]}} [P_{B_j}]. \quad (7)$$

The translation in step  $k$  is given as a rotation from the previous step,  $\Delta \mathbf{t} = g_{[P_B]} - R \cdot g_{P_A}$ . Assume that total number of iterations is  $K$ ; then, the final translation vector is

$$\mathbf{t} = \mathbf{t}_{K-1} + \Delta \mathbf{t}. \quad (8)$$

To calculate the rotation matrix  $R$ , all the points must be first moved to their centroids and formulated as

$$p'_{A_i} = P_{A_i} + g_{P_A}$$

$$[p'_{B_j}] = [P_{B_j}] + g_{[P_B]}. \quad (9)$$

According to [29], the incremental rotation matrix  $\Delta R$  is computed by

$$\Delta R = VU^T, \quad (10)$$

where  $V$  and  $U$  are orthogonal matrices that result from a singular value decomposition where

$$W = U \sum V^T = \sum_{i=1}^{N_{P_A}} [p'_{B_i}] (p'_{A_i})^T. \quad (11)$$

Here,  $\Delta R$  is the orientation increment between the matching points in  $[p_{B_j}]$  and  $p_{A_i}$ , which stem from  $[P_B]$  and  $P_A$ , respectively. The final rotation matrix  $R$  is formulated as follows:

$$R = R_{K-1} + \Delta R. \quad (12)$$

ICP aligns the two point clouds by solving a least squares problem [30] with an error function  $\sigma(R, \mathbf{t})$  formulated as

$$\sigma(R, \mathbf{t}) = \frac{1}{N_{P_A}} \sum_{i=1}^{N_{P_A}} w_i \left\| [P_{B_j}] - R \cdot p_{A_i} - \mathbf{t} \right\|^2, \quad (13)$$

where  $N_{P_A}$  represents the number of points in  $P_A$ . Points that do not meet the error metric (such as a set Euclidean distance between  $p_{A_i}$  and  $[p_{B_j}]$  for outlier removal) have a weight of 0 so that they do not contribute to the error function. The termination criterion  $[\sigma]$  is checked after all the iteration steps have been processed. The iteration continues until the termination criterion  $\Delta\sigma = \sigma_k - \sigma_{k-1} < [\sigma]$  is reached in step  $k$ , meaning that the error delta between two subsequent frames is below the termination criterion  $[\sigma]$ . The resulting translation vector  $\mathbf{t}$  and the rotation matrix  $R$  are generated in step  $k$ .

(3) *Solution of the Modified Point Cloud Set.* After transforming the point cloud set from  $[P_B]$  to the position registered with  $P_A$ , a new point cloud set  $\overline{P_B}$  is generated. The modified point cloud set  $P^*$ , for arbitrary  $p_i^* \in P^*$ ,  $p_i^*$  is computed as shown in the following:

$$p_i^* = \frac{(p_{A_i} + \overline{p_{B_j}})}{2}. \quad (14)$$

3.3. *Improved Poisson Surface Reconstruction Algorithm.* Surface reconstruction can be accomplished by a Poisson surface reconstruction algorithm. Typical PDE Poisson equations are constructed and solved iteratively to extract the isosurface and realize surface reconstruction. However, there are several disadvantages to this algorithm: it is strongly dependent on normal vectors and tends to form mesh holes. Therefore, the IPSR is proposed in this paper. The point cloud pre-treatment, including recomputing the normal vectors based on the least squares method and postprocessing a generated PDE patch based on biharmonic-like fourth-order PDEs, is conducted successively to generate an accurate, connected, and lightweight mesh.

3.3.1. *Recomputation of Point Clouds Normal Vectors.* The recomputation of the normal vectors of a point cloud proposed in this paper is based on the least squares method. Then, the computational complexity of the surface reconstruction algorithm is reduced based on the accuracy of the normal vectors [31].

An arbitrary coordinate point  $p_i^*$ , in a point cloud set  $P^*$ , and all the  $K$  neighbor points of  $p_i^*$  can be regarded as being

on approximately the same plane,  $L$ , which is formulated as follows:

$$Ax + By + Cz + D = 0. \quad (15)$$

The residual sum of squares between all the  $K$ -neighbor points and  $L$  is

$$d = \sum_{i=1}^n (Ax_i^* + By_i^* + C - z)^2. \quad (16)$$

The best fit can be regarded as the minimum  $d$ ; therefore, we take the partial derivative of  $d$  with  $A$ ,  $B$ , and  $C$ . Then, the equations are

$$\begin{aligned} \frac{\partial d}{\partial A} &= 2 \left( A \sum_{i=1}^n x_i^2 + B \sum_{i=1}^n x_i y_i + C \sum_{i=1}^n x_i - \sum_{i=1}^n x_i z_i \right) \\ \frac{\partial d}{\partial B} &= 2 \left( A \sum_{i=1}^n x_i y_i + B \sum_{i=1}^n y_i^2 + C \sum_{i=1}^n y_i - \sum_{i=1}^n y_i z_i \right) \\ \frac{\partial d}{\partial C} &= 2 \left( A \sum_{i=1}^n x_i + B \sum_{i=1}^n y_i + nC - \sum_{i=1}^n z_i \right). \end{aligned} \quad (17)$$

Let  $\partial d / \partial A = 0$ ,  $\partial d / \partial B = 0$ , and  $\partial d / \partial C = 0$ . Finally, the result can be formulated as follows:

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n x_i^{*2} & \sum_{i=1}^n x_i^* y_i^* & \sum_{i=1}^n x_i^* \\ \sum_{i=1}^n x_i^* y_i^* & \sum_{i=1}^n y_i^{*2} & \sum_{i=1}^n y_i^* \\ \sum_{i=1}^n x_i^* & \sum_{i=1}^n y_i^* & n \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^n x_i^* z_i^* \\ \sum_{i=1}^n y_i^* z_i^* \\ \sum_{i=1}^n z_i^* \end{bmatrix}. \quad (18)$$

The coefficients  $A$ ,  $B$ , and  $C$  can be solved via (18) to generate the normal vector  $n_i$  of  $p_i^*$ . Moreover, the vector orientations are processed for consistency; that is, an arbitrary normal vector  $n_j$  of point  $p_j^*$ , which is near point  $p_i^*$ , is selected. Transvection of  $n_i$  and  $n_j$  is conducted. If  $n_i \cdot n_j > 0$ , the orientation of  $n_i$  is kept; otherwise,  $n_i$  is adjusted to the opposite orientation:  $-n_i$ .

3.3.2. *Poisson Surface Reconstruction.* In a Poisson surface reconstruction, the solution of the surface to be reconstructed is regarded as the solution of a 3D indicator function  $\chi$ . The reconstructed surface is then obtained by extracting the appropriate isosurface [32]. This process is depicted in Figure 4.

To solve the Poisson equation, the 3D space should be discretized. Therefore, an adaptive *octree* is built. The closer the border is, the larger the density of the *octree* is [32]. Based on the *octree*, the base function  $F$  is defined as

$$F(x, y, z) \equiv (B(x) B(y) B(z))^n \quad \text{with } B(t) = \begin{cases} 1 & |t| < 0.5 \\ 0 & \text{otherwise,} \end{cases} \quad (19)$$

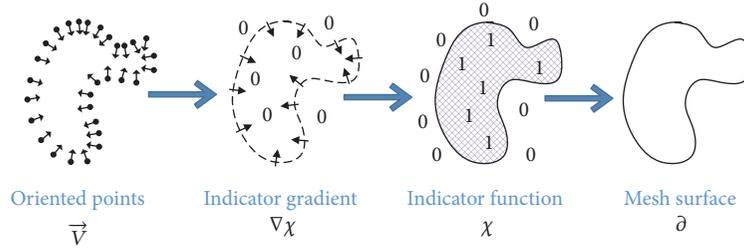


FIGURE 4: The process of Poisson surface reconstruction.

where  $*n$  represents the  $n$ th convolution. Every point sample becomes a leaf node  $o$  in a depth  $D$  *octree*; therefore, the unit-integral “node function” centered around node  $o$  is formulated as

$$F_o(q) = F\left(\frac{q - o \cdot c}{o \cdot w}\right) \frac{1}{(o \cdot w)^3}, \quad (20)$$

where  $o \cdot c$  represents the center of  $o$  and  $o \cdot w$  represents the width of  $o$ .

The vector field  $\vec{V}(q)$  is determined from the normal vector  $\vec{n}_i$  of  $p_i^*$  in a point cloud set  $P^*$ . The gradient field of the indicator function  $\chi$  can be expressed as  $\nabla\chi$ . The purpose of the Poisson surface reconstruction is to solve a function  $\chi$  whose gradient best approximates a vector field  $\vec{V}(q)$  defined by  $P^*$ . This can be formulated as follows:

$$\min_{\chi} \|\nabla\chi - \vec{V}\|. \quad (21)$$

The problem shown in (21) can be solved by the Poisson equation  $\Delta\chi = \nabla \cdot \vec{V}$ . A vector field  $\vec{V}$  can be approximately formulated as

$$\vec{V}(q) = \sum_{p^* \in P^*} \sum_{o \in Ngr_D(p^*)} \alpha_{o,s} F_o(q) \vec{n}, \quad (22)$$

where  $Ngr_D(s)$  are the eight depth- $D$  nodes closest to  $p^*$  and  $\{\alpha_{o,s}\}$  represents the trilinear interpolation weights. To solve the problem that  $\nabla \cdot \vec{V}$  and  $\Delta\chi$  are not in the function space, the equation can be simplified by solving for the  $\chi$  function by minimizing

$$\begin{aligned} & \sum_o \|\langle \Delta\chi - \nabla \cdot \vec{V}, F_o \rangle\|^2 \\ & = \sum_o \|\langle \Delta\chi, F_o \rangle - \langle \nabla \cdot \vec{V}, F_o \rangle\|^2. \end{aligned} \quad (23)$$

Given a vector  $v$ , whose  $o$ th coordinate is  $\vec{v}_o = \langle \nabla \cdot \vec{V}, F_o \rangle$ , the goal is to solve for the function  $\chi$ . The vector can be obtained by projecting the Laplacian of  $\chi$  onto each  $F_o$ , ensuring that it is as close to  $v$  as possible. To express

(23) in matrix form, let  $\chi = \sum_o x_o F_o$ ; then, the problem is transformed into one of solving for the vector  $x_o$ .

$$\begin{aligned} & \sum_o \|\langle \Delta\chi, F_o \rangle - \langle \nabla \cdot \vec{V}, F_o \rangle\|^2 \\ & = \sum_{o'} \left\| \sum_o x_o \langle \Delta F_o, F_{o'} \rangle - \langle \nabla \cdot \vec{V}, F_{o'} \rangle \right\|^2. \end{aligned} \quad (24)$$

Let the matrix  $L$  be contributed as the dot product of the Laplacian with each of the  $F_o$  values; for all  $o, o'$ ; the  $(o, o')$ th entry of  $L$  is set to

$$\begin{aligned} L_{o,o'} & \equiv \left\langle \frac{\partial^2 F_o}{\partial x^2}, F_{o'} \right\rangle + \left\langle \frac{\partial^2 F_o}{\partial y^2}, F_{o'} \right\rangle \\ & + \left\langle \frac{\partial^2 F_o}{\partial z^2}, F_{o'} \right\rangle. \end{aligned} \quad (25)$$

Thus, the solution of the indicator function  $\chi$  can be regarded as finding

$$\min_x \|Lx - v\|^2. \quad (26)$$

The isosurface from the indicator function  $\chi$  can be extracted to *octree* representations through the method used in previous adaptations of the marching cubes [33].

**3.3.3. Generation of PDE Patches.** The lightweight methods are implemented in the abovementioned algorithms (i.e., PCU in Section 3.1, RICP in Section 3.2, and the recomputation of point clouds normal vectors in Section 3.3.1). Because of the highly lightweight point cloud data, it is easy to generate mesh holes near areas with complex curvature variations, which could influence the visualization of the reconstructed 3D model. Thus, in this paper, an alternative approach to PDE patch generation based on biharmonic-like fourth-order PDEs is proposed. The algorithm is utilized to repair mesh holes on the reconstructed surface according to Section 3.3.2. Each hole is represented by patches with their own  $uv$  coordinate system. The preservation of irregular and sharp details on the surface will be found by matching the respective patches to the surface at various sizes and orientations. According to [34], a biharmonic-like fourth-order PDE is formulated as follows:

$$\left( \frac{\partial^2}{\partial u^2} + a^2 \frac{\partial^2}{\partial v^2} \right)^2 S(u, v) = 0, \quad (27)$$

where  $S(u, v)$  is regarded as the hole  $H(u, v)$  to be repaired, and  $0 \leq u \leq 1$ ,  $0 \leq v \leq 2\pi$ . Additionally, the PDE patch  $P(u, v)$  can be formulated by (27). The analytic solution of (27) can be computed by of separating the variables and formulating the problem as

$$H(u, v) = A_0(u) + \sum_{n=1}^N [A_n(u) \cos(nv) + B_n(u) \sin(nv)], \quad (28)$$

where  $A_0(u)$ ,  $A_n(u)$ ,  $B_n(u)$  are, respectively, formulated as follows:

$$\begin{aligned} A_0(u) &= \alpha_{00} + \alpha_{01}u + \alpha_{02}u^2 + \alpha_{03}u^3 \\ A_n(u) &= \alpha_{n1}e^{anu} + \alpha_{n2}ue^{anu} + \alpha_{n3}e^{-anu} + \alpha_{n4}ue^{-anu} \\ B_n(u) &= \beta_{n1}e^{anu} + \beta_{n2}ue^{anu} + \beta_{n3}e^{-anu} + \beta_{n4}ue^{-anu}. \end{aligned} \quad (29)$$

The PDE coefficients  $\alpha_{00}, \alpha_{01}, \dots, \alpha_{n3}, \alpha_{n4}$  and  $\beta_{11}, \beta_{12}, \dots, \beta_{n3}, \beta_{n4}$  are vectors valued by the boundary conditions.  $A_0(u)$  is regarded as the ‘‘spine’’ of the surface, and  $A_n(u) \cos(nv) + B_n(u) \sin(nv)$  is an  $n$ -order ‘‘radius’’ vector. The amplitude of the ‘‘radius’’ term decays as the frequency increases (if  $N \rightarrow \infty$ ), so that the solution of the PDE patch is mostly determined by a first-order ‘‘radius’’ vector  $A_1(u) \cos(v) + B_1(u) \sin(v)$  and a second-order ‘‘radius’’ vector  $A_2(u) \cos(2v) + B_2(u) \sin(2v)$ . Therefore, (28) can be approximately reformulated as follows:

$$\begin{aligned} H(u, v) &= A_0(u) + A_1(u) \cos(v) + B_1(u) \sin(v) \\ &\quad + A_2(u) \cos(2v) + B_2(u) \sin(2v). \end{aligned} \quad (30)$$

The number of vector-valued PDE coefficients is  $M_c = 4 \times (2N + 1)$ . The higher  $M_c$  is, the higher the complexity of the reconstruction is. To generate a lightweight mesh, we let  $N = 3$ . Then, the boundary conditions that fit (27) are formulated as follows:

$$\begin{aligned} H(0, v) &= C_0(v) \\ H(u_1, v) &= C_1(v) \\ H(u_2, v) &= C_2(v) \\ H(1, v) &= C_3(v), \end{aligned} \quad (31)$$

where the conditions  $C_0(v)$ ,  $C_1(v)$ ,  $C_2(v)$ , and  $C_3(v)$  are isoparm boundary curves on the surface patch at  $u = 0, u_1, u_2, 1$ , respectively, where  $0 < u_1, u_2 < 1$ . To precisely approach the mesh representation of complex geometric shapes, this method needs to use sufficient PDE boundary curves, which must be extracted from the vertices of the original polygon mesh representation. Figure 5 illustrates the layout of the various PDE boundary curves for an individual PDE patch, where  $C_0(v)$  degenerates into a single point.

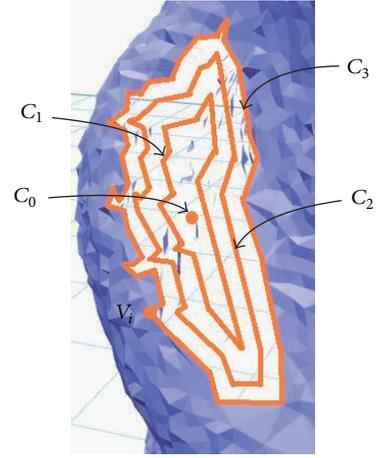


FIGURE 5: Layout of PDE boundary curves.

## 4. Experiment and Analysis

**4.1. System Framework and Establishment of Experimental Platform.** To verify the feasibility of the proposed lightweight surface reconstruction method proposed in this paper, an online personalized customization system oriented toward 3D printing was developed and used as the experimental environment. The system framework is shown in Figure 6 and can be divided into 4 units: Central Control Unit; Online Personalized Customization Unit; 3D Scanning Unit; and 3D Printing Unit. The Central Control Unit runs on a PC and a cloud server, and the collaborative operation of the other 3 units is controlled by the Central Control Unit, which is the core of the whole system. The Online Personalized Customization Unit runs in Google Chrome directly. The dynamic visualization of the point cloud generated by the 3D Scanning Unit can be rendered; that is, the point cloud generated in real-time can be displayed to users. Moreover, the point clouds are reconstructed as meshes. Modifications of color, texture, dimension, slicing of the STL model, and generation of G codes are also supported in the Online Personalized Customization Unit. G codes can be imported by the 3D Printing Unit for the 3D printing tasks.

**(1) Hardware Framework of 3D Scanning Unit.** The 3D Scanning Unit is composed of the open source 3D scanner Ciclop, whose hardware framework is shown in Figure 7. The Ciclop is composed of an Arduino UNO Motherboard (control core), a Logitech C270 (camera), two infrared laser modules, and other mechanical structures. Its turn table is driven by a NEMA 17 stepper motor which rotates the object undergoing scanning at  $1.8^\circ/s$  around its central axis. Simultaneously, two sets of 5 mW infrared laser modules, set at an angle of  $135^\circ$ , are coordinated with the Logitech C270 to generate depth data at different angles. Thus, the point cloud data of the scanned object can be generated as well [35].

**(2) Dynamic Visualization of Point Cloud Data.** To achieve the dynamic visualization, the point cloud data from the 3D Scanning Unit to the Online Personalized Customization

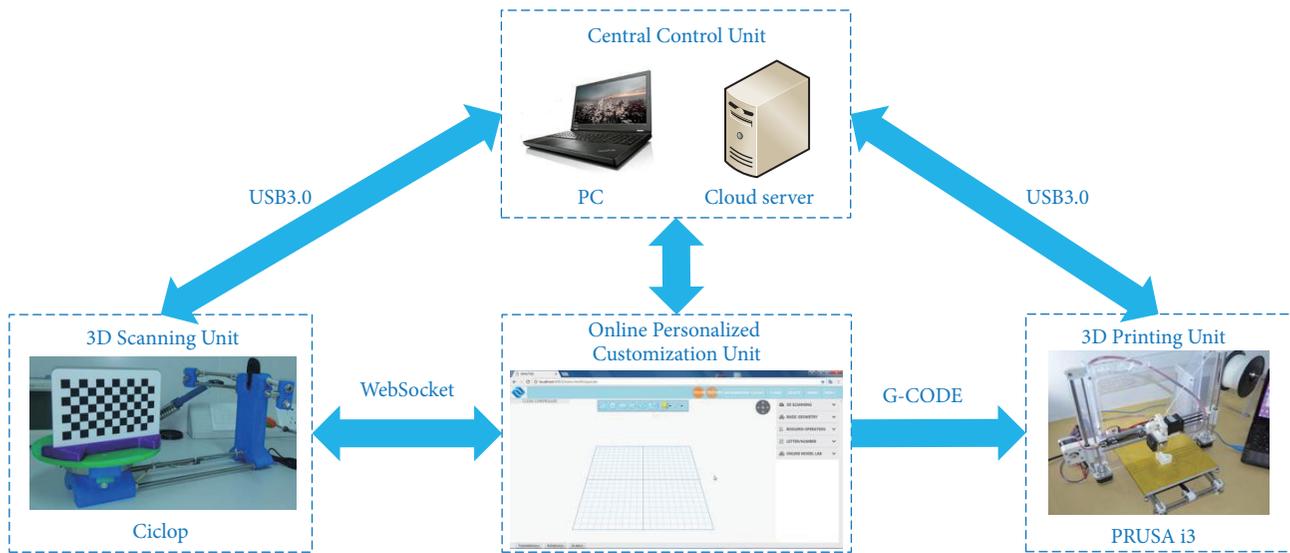


FIGURE 6: Framework of the online personalized customization system oriented toward 3D printing.

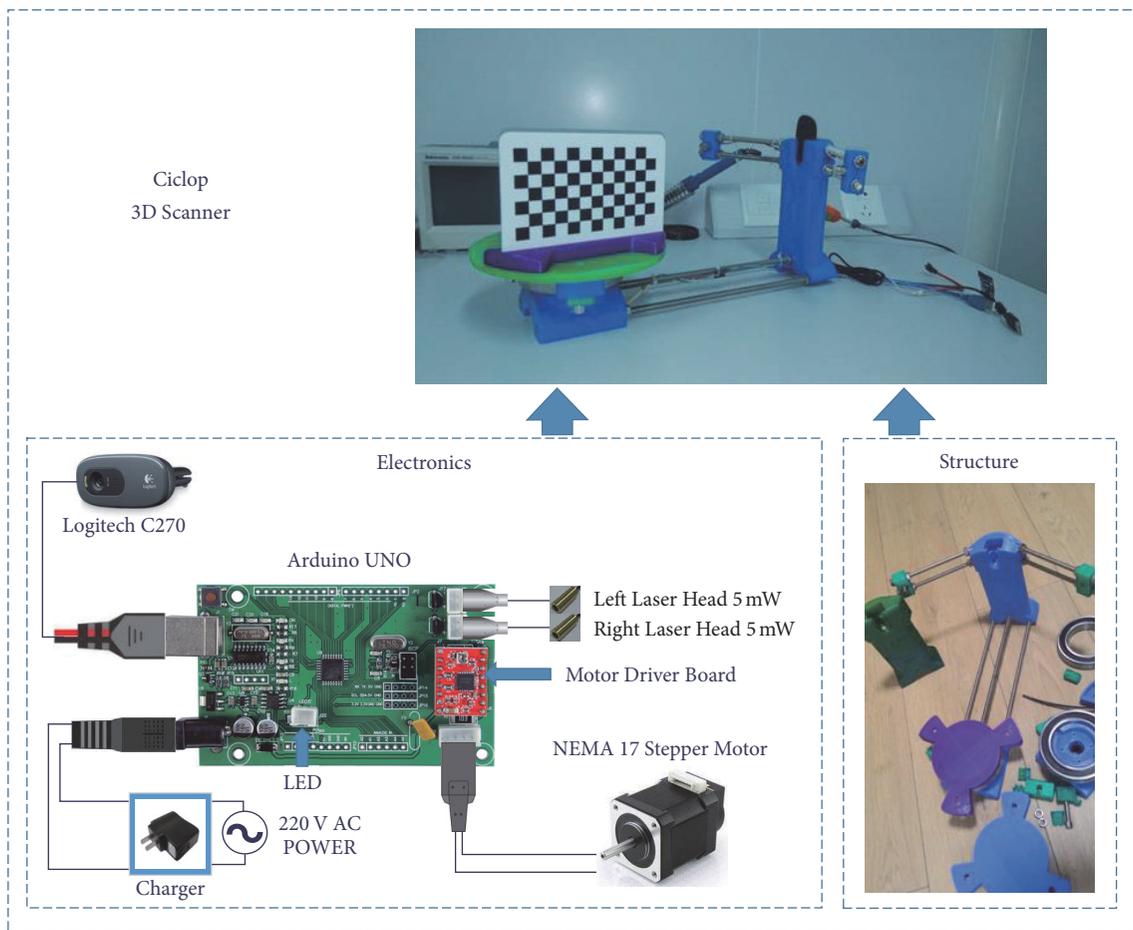


FIGURE 7: Hardware framework of 3D Scanning Unit.

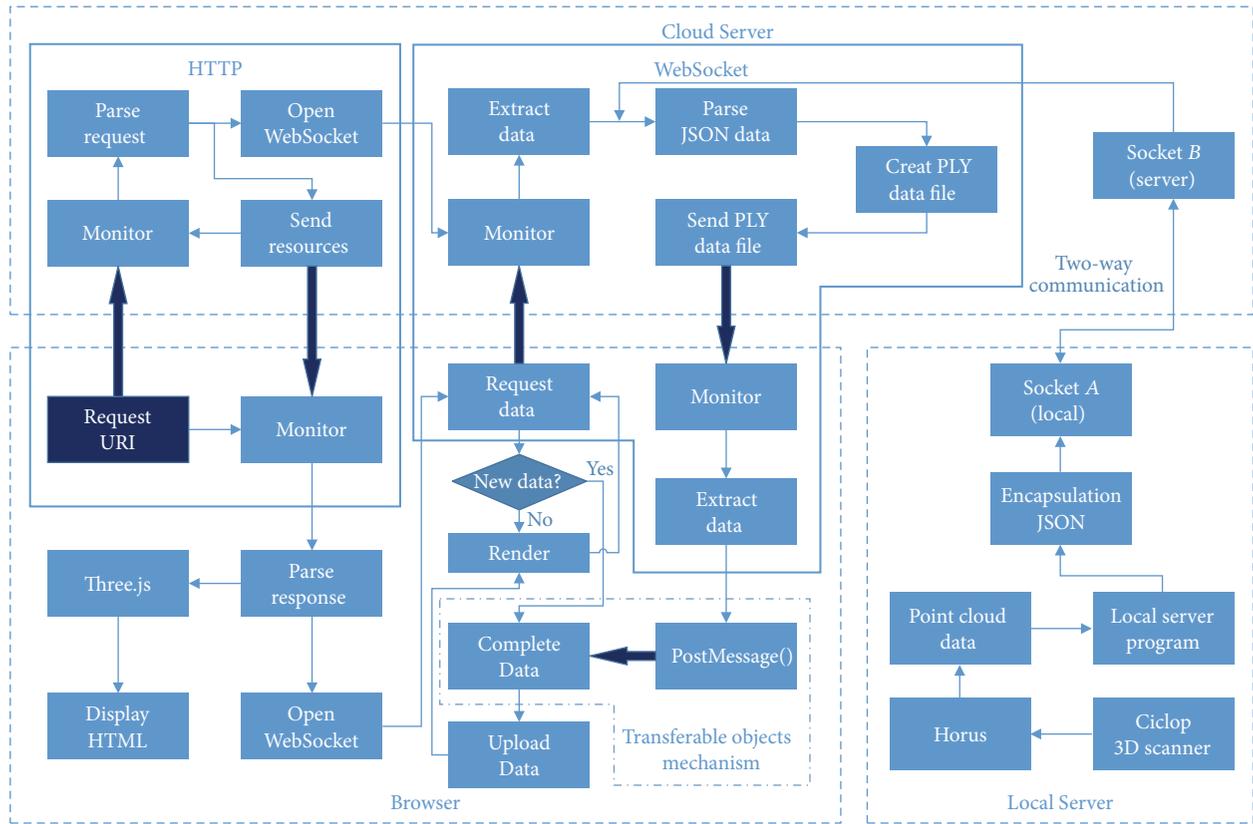


FIGURE 8: Dynamic point cloud visualization framework.

Unit needs to be transmitted in real time. A proposed dynamic visualization framework for point clouds based on WebSocket is shown in Figure 8. The thin arrows represent the control flow within the server or browser, while the wide arrows represent the data flow between memory spaces. WebSocket is a real-time communication protocol for HTML5 that can achieve real-time data transmission between a browser and a cloud server [36]. Thus, the point cloud data that are generated by Ciclop in real time can be dynamically displayed in the browser by WebSocket and WebGL. The framework is divided into three parts: cloud server, local server, and browser.

The interaction is initiated by the browser, which sends a request URI to the cloud server. The cloud server parses the request and sends the appropriate JavaScript code back to the browser, which dynamically visualizes and renders the point cloud. Meanwhile, a request is also sent to Ciclop. The point clouds generated by Ciclop are extracted by its underlying computer software, Horus, and encapsulated as JSON files by the local server. A socket connection can be established between a local server and cloud server to upload JSON files. Subsequently, a WebSocket connection will also be established through JavaScript code running in the Browser to fetch and parse the JSON files. After the request URI is sent from the browser to the cloud server, a monitor is established to determine whether the latest JSON file to be parsed in the browser through WebSocket connection exists. The point

cloud data will be dynamically updated by WebGL when the JSON file exists.

JavaScript is generally single-threaded; consequently, the performance requirements of the computer hardware are higher. To allow the user-interface to remain responsive, the browser fetches and renders the JSON files. The fetching operation runs in a separate Web Worker thread, while the rendering runs in the main JavaScript thread. All long-running code paths need to run outside the main thread [37]. Moreover, to avoid browser congestion by too many messages, the next time step of a JSON file is always requested explicitly by the browser. That is, only when the browser acquires a new rendering request for the current point cloud data to GPU does it send the cloud server a new request for new JSON files. Thus, this framework ensures that each frame will be rendered at least once.

(3) *Establishment of the Experimental Platform.* The hardware configuration of the Central Control Unit is shown in Table 1.

The environmental calibration parameters of the 3D scanning operation to meet the requirements of the experimental environment are configured in Horus, the computer software underlying Ciclop, as shown in Table 2.

4.2. *Point Cloud Update and Dynamic Visualization Experiments.* Using the dynamic point cloud visualization

TABLE 1: Hardware configuration of Central Control Unit.

Hardware	PC	Cloud server
CPU	Intel Core i7 7700K 4.2 GHz	Intel Skylake Xeon Platinum 8163 2.5 GHz
GPU	ASUS RX550 4 GB	ASUS NVIDIA GeForce GTX1080 8 GB
RAM	8 GB	10 GB
Hard Disk	SSD 128 GB	SSD 40 GB
Bandwidth	10 Mbps	50 Mbps

TABLE 2: Environmental calibration parameters of experiments.

Indexes	Brightness	Contrast	Saturation	Exposure	Step degrees	Photograph interval
Value	150	34	32	10	0.45°	0.5 s

TABLE 3: Generation time and amount of data on two sets of point clouds.

Model type	Dynamic visualization with PCU		Dynamic visualization without PCU	
	Load time/s	Size/MB	Load time/s	Size/MB
Huba	109	1.60	94	5.19
Totoro	126	1.89	112	6.54

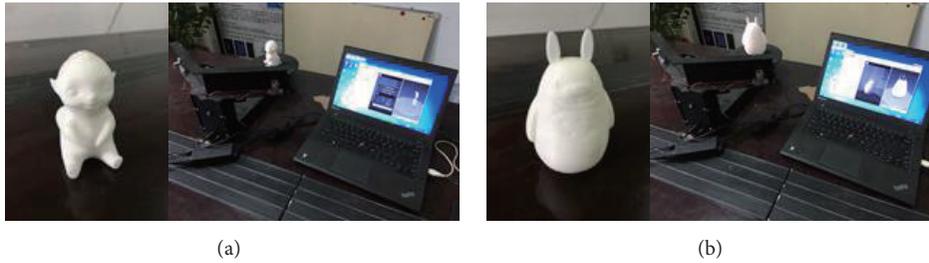


FIGURE 9: Experimental environments of 3D scanning.

framework and the PCU as proposed in Section 3.1, experiments assessing the online 3D scanning were performed on the “Huba” and “Totoro” models to verify the effect of point cloud dynamic visualization and the lightweight effect of the PCU. The experimental environment is shown in Figure 9, and the dynamic visualization processes in Google Chrome are shown in Figure 10. To optimize the visualization effects, the background color of Google Chrome was set to black.

The generation time and amount of data of the two sets of point clouds (representing the two models “Huba” and “Totoro”) were recorded. For comparison purposes, the two circumstances were implemented both with and without the PCU. The results are shown in Table 3.

As listed in Table 3, with the PCU, the size of the point cloud data of the “Huba” model decreases by 69.2% (from 5.19 MB to 1.60 MB), while the size of the point cloud data of the “Totoro” model decreases by 71.1% (from 6.54 MB to 1.89 MB) during the point cloud dynamic visualization process. Although the load time of each model increases by approximately 15% with the PCU, the growth is not substantial compared to the lightweight point cloud data. Experimental analysis shows that lightweight dynamic visualization of point cloud data in a web environment

can be accomplished by the PCU in conjunction with the dynamic visualization framework for point clouds based on WebSocket as proposed in this paper.

**4.3. Point Cloud Registration Experiments.** Experiments on the registration of two sets of point clouds,  $P_A$  and  $P_B$ , which belong to the “Huba” and “Totoro” models, were conducted by RICP and ICP, respectively, to generate a modified point cloud set  $P^*$ . The termination criteria of the MSE threshold was set to  $[\sigma] = 0.005$  mm, and the number of iterations and the registration time of the two point cloud sets by RICP and ICP were recorded and are listed in Table 4, and the convergence curves of the registration error with the two sets of point clouds are shown in Figure 11.

The registration effects of the two sets of point clouds calculated by the different algorithms are shown in Figure 12.

With the RICP, the number of iterations decreased from 12 to 7, and the time decreased by 45.8% from 59 s to 32 s for the “Huba” model, while the number of iterations decreased from 15 to 8, and the time decreased by 46.8% from 77 s to 41 s for the “Totoro” model. These results are listed in Table 4. Figure 11 shows that the MSEs of the two models decrease by 0.039 mm and 0.036 mm after one iteration.

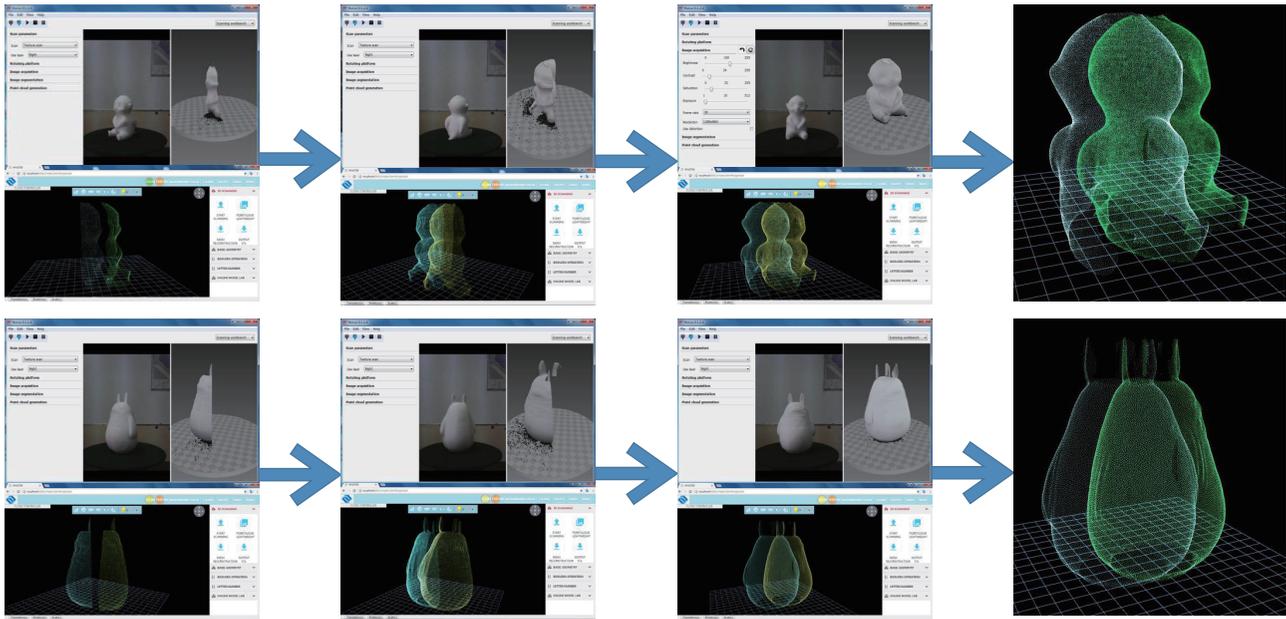


FIGURE 10: Dynamic visualization processes of point cloud.

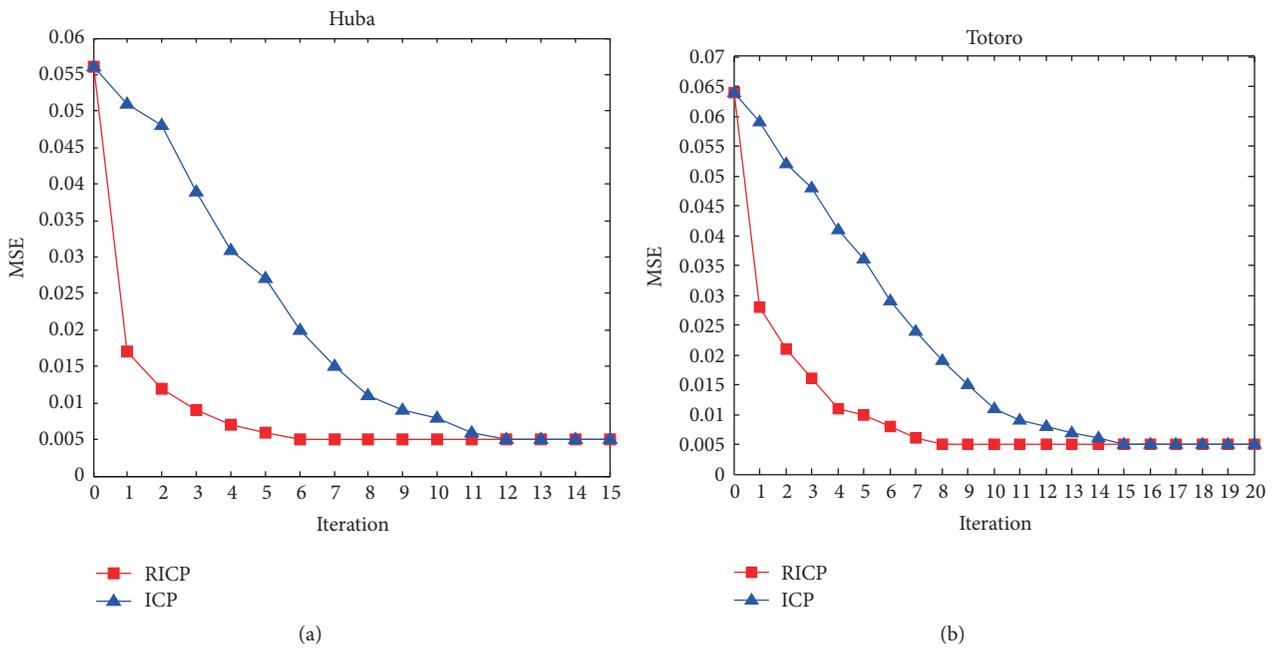


FIGURE 11: Convergence curves of the registration error.

TABLE 4: Number of iterations and elapsed time of registration by RICP and ICP.

Registration algorithm	[ $\sigma$ ]/mm	Huba		Totoro	
		Iteration	Time/sec	Iteration	Time/sec
RICP	0.005	7	32	8	41
ICP		12	59	15	77

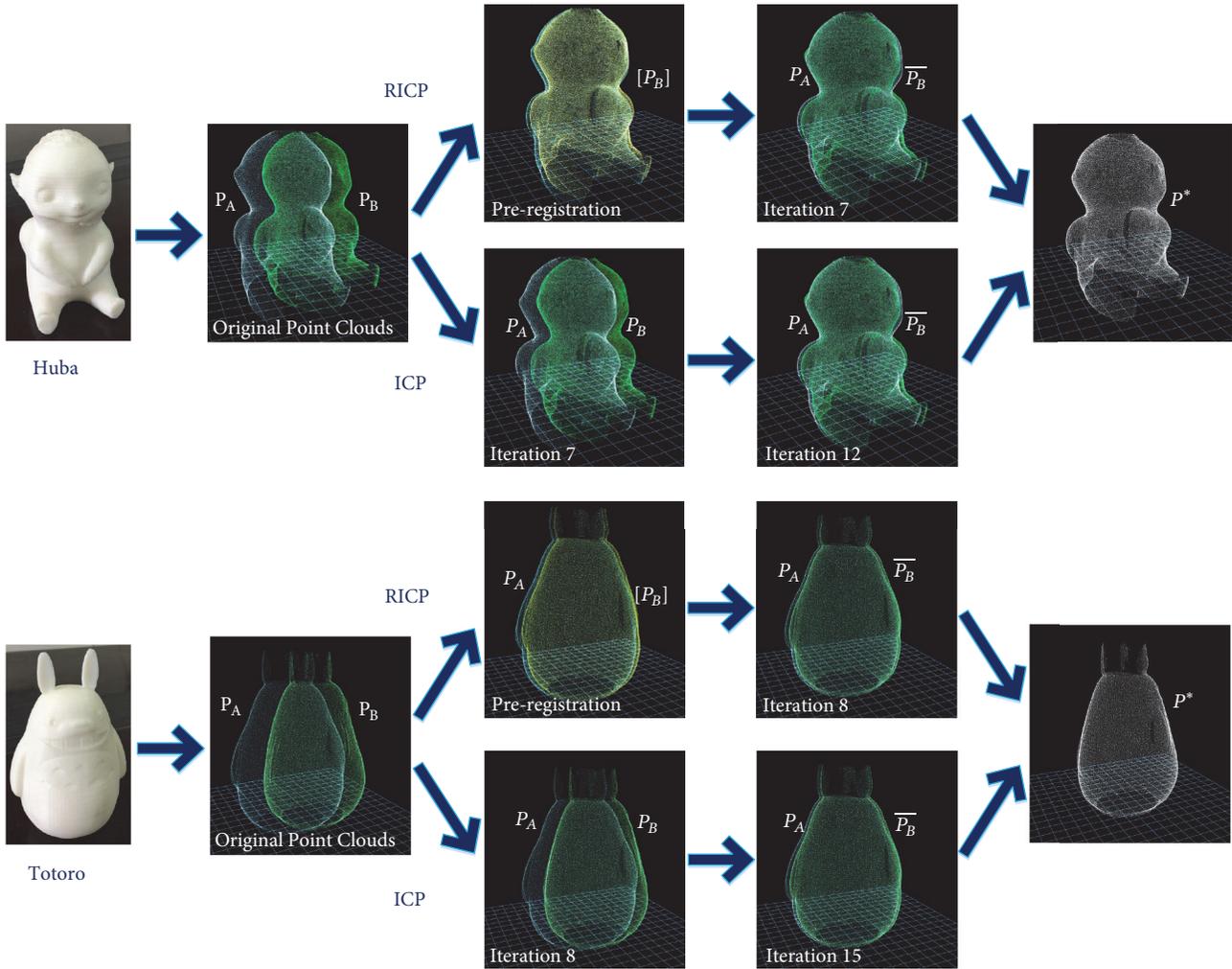


FIGURE 12: Visualization effect of registration by RICP and ICP.

TABLE 5: Relevant indicators of surface reconstruction by IPSR and PSR.

Model	Points	Algorithm type	Triangular facets	Size/MB (STL)	Reconstruction time/s	Average FPS	Occupied CPU
Huba	85432	PSR	73735	14.4	3.4	62	8%
		IPSR	61476	10.5	2.9	61	7%
Totoro	129463	PSR	101950	20.1	7.9	76	10%
		IPSR	89759	17.2	6.5	74	8%

The experimental analysis shows the increase in algorithmic efficiency from preregistering the point cloud data in the RICP. The visualization effects of the registrations by the RICP and ICP are shown in Figure 12.

4.4. *Online Surface Reconstruction Experiments.* Using the IPSR proposed in Section 3.3, modified point cloud data sets of the “Huba” and “Totoro” were reconstructed by both IPSR and Poisson surface reconstruction (PSR) algorithm in the web environment. The online surface reconstructions by both algorithms are shown in Figure 13.

Relevant indicator values were recorded during the process of surface reconstruction to assess IPSR and PSR, including triangular facets, size, reconstruction time, average frames per second (FPS), and occupied CPU percentage [38]. The specific values of these metrics are listed in Table 5, while four histograms of the triangular facets, size, reconstruction time, and average FPS are shown in Figure 14.

As Figure 13 shows, several single-point spikes and holes exist on the polygon mesh reconstructed by PSR. In contrast, the polygon meshes reconstructed by IPSR are accurate, connected, and lightweight. In fact, the surface is relatively

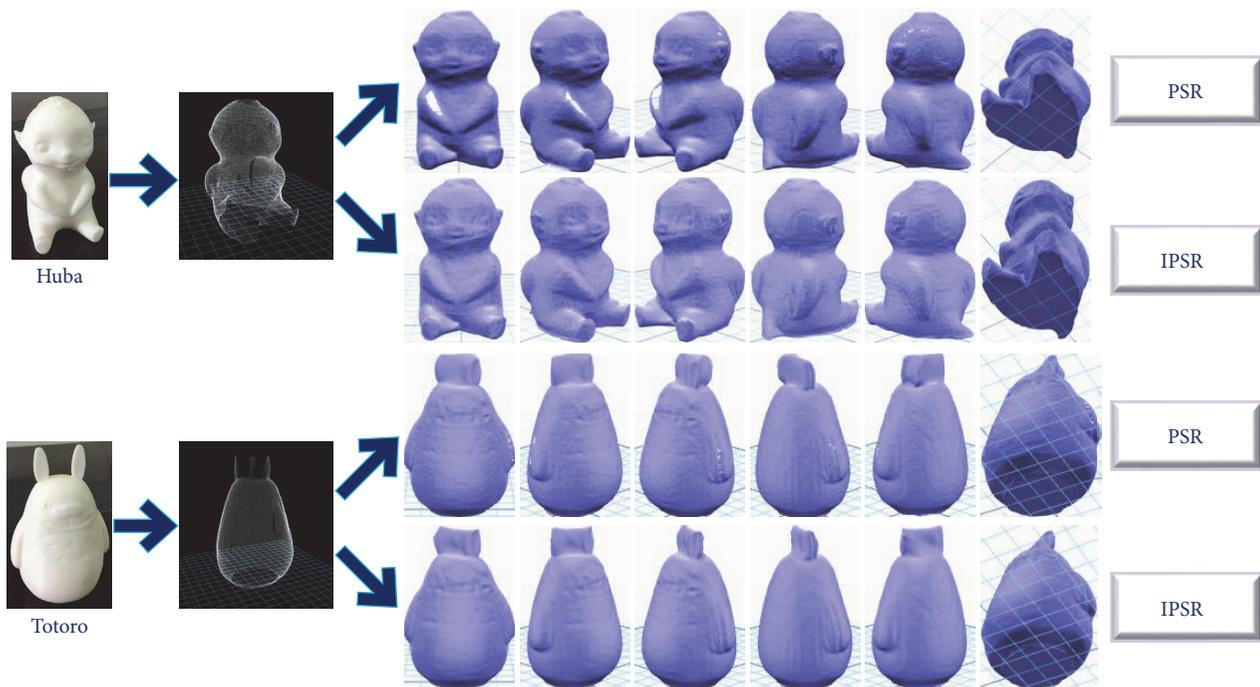


FIGURE 13: Online surface reconstruction effect by IPSR and PSR.

smooth, without any obvious single-point spikes on the hole boundaries.

In addition to the differences in Table 5 and Figure 14, compared with PSR, the triangular facets of the “Huba” mesh created by IPSR decreased by 16.6%, while the size and reconstruction time decreased by 27.1% and 14.7%, respectively. The triangular facets of the “Totoro” mesh decreased by 12.1%, while the size and reconstruction time decreased by 14.4% and 17.7%, respectively. The lightweight extent of each mesh produced by IPSR is relatively obvious. Although the decrease in average FPS is barely noticeable (1.6% and 2.6% for the “Huba” mesh and “Totoro” mesh, resp.) both are still relatively acceptable with tiny fluctuations around a median value of approximately 70. Users reported that the system ran smoothly without any obvious decrease in rendering speed.

The experimental analysis shows that the reconstruction efficiency, the accuracy, and the absence of obvious spikes in the mesh reconstructed by IPSR are more desirable given the lightweight data requirements of a web environment. Additionally, meshes on a 3D printing cloud server platform can be personalized and customized using this method.

## 5. Conclusions

In this paper, a lightweight surface reconstruction method for online 3D scanning point cloud data oriented toward 3D printing is proposed. To achieve low algorithmic complexity and generate a lightweight 3D model, an online lightweight surface reconstruction algorithm is proposed, which is composed of a point cloud update algorithm (PCU), a rapid iterative closest point algorithm (RICP), and the improved

Poisson surface reconstruction algorithm (IPSR). The PCU is used to denoise the point cloud data to create a lightweight version in real time. The RICP is used to perform rapid and accurate registration between two sets of point cloud data. The IPSR is used to generate the lightweight mesh, and the postprocessing of the PDE patch generation based on biharmonic-like fourth-order PDEs is executed to repair the mesh holes on the reconstructed lightweight mesh, which improves the 3D model visualization. In addition, to achieve real-time point cloud data transmission in conjunction with the 3D scanning process in a web environment, a dynamic visualization framework for point cloud data based on Web-Socket is also proposed. This approach achieves comprehensive dynamic visualization of the point cloud data in the browser. The fluency and high rendering effect are ensured through the Web Worker mechanism in the high concurrency environment. Finally, an online personalized customization system oriented toward 3D printing is developed based on the proposed method.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work is supported by the National Key Technology Research and Development Program of China (no. 2016YFB1101700) and Nature Science Foundation of Hubei Province, China (no. 2015CFA115).

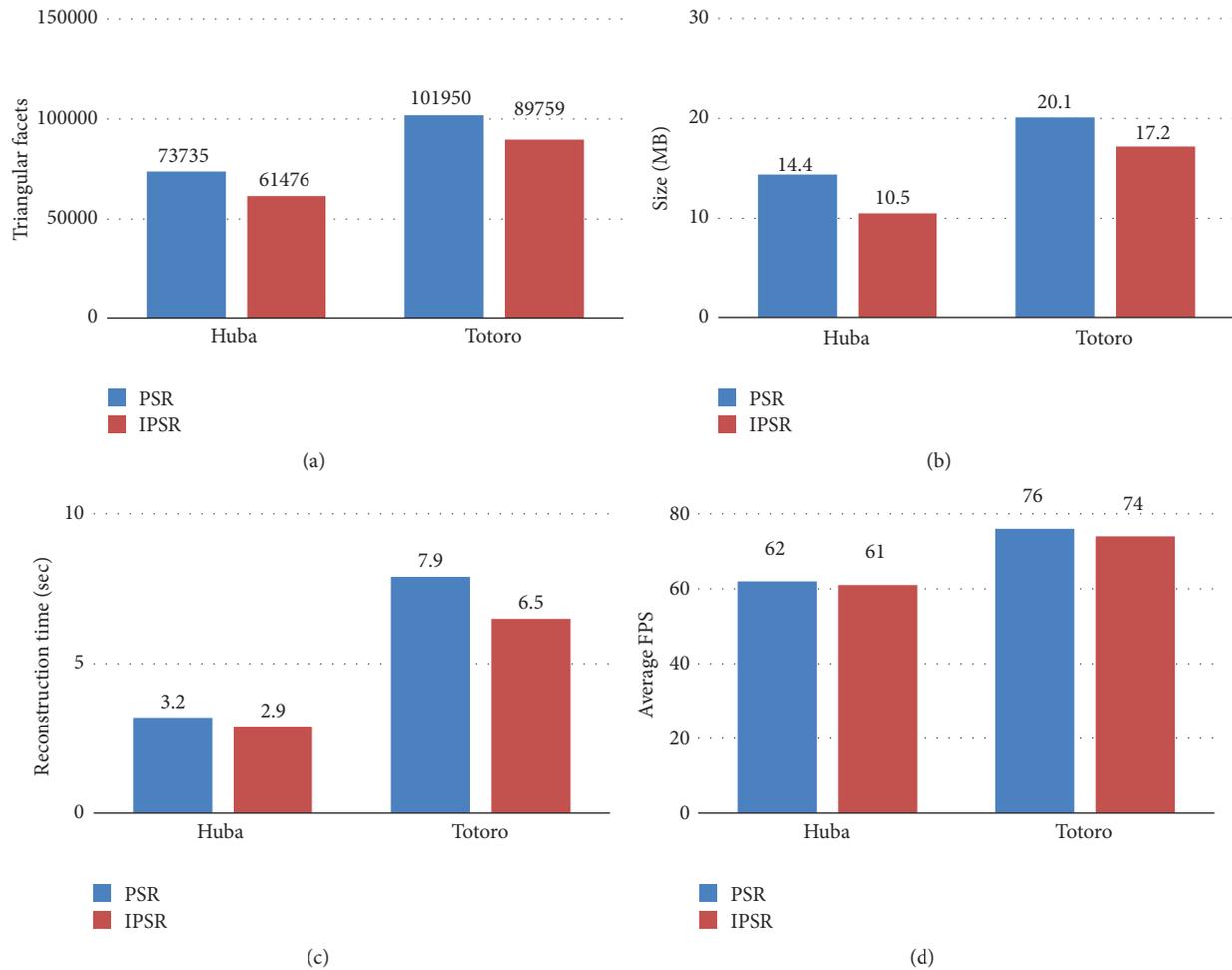


FIGURE 14: Histograms of relevant indicators.

## References

- [1] J. Klein, M. Stern, G. Franchin et al., "Additive Manufacturing of Optically Transparent Glass," *3D Printing and Additive Manufacturing*, vol. 2, no. 3, pp. 92–105, 2015.
- [2] P. Markillie, "A third industrial revolution," *The economist*, vol. 21, pp. 3–12, 2012.
- [3] X. Yao and Y. Lin, "Emerging manufacturing paradigm shifts for the incoming industrial revolution," *The International Journal of Advanced Manufacturing Technology*, vol. 85, no. 5-8, pp. 1665–1676, 2016.
- [4] W. Oropallo and L. A. Piegl, "Ten challenges in 3D printing," *Engineering with Computers*, vol. 32, no. 1, pp. 135–148, 2016.
- [5] C. Lim, "The Implementation of 3D Printing in Customized Interactive Design for Elderly Welfare Technology," in *HCI International 2014 - Posters' Extended Abstracts*, vol. 435 of *Communications in Computer and Information Science*, pp. 299–303, Springer International Publishing, Cham, 2014.
- [6] J. Mai, L. Zhang, F. Tao, and L. Ren, "Customized production based on distributed 3D printing services in cloud manufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 84, no. 1-4, pp. 71–83, 2016.
- [7] J. Xu, L. Ding, and P. E. D. Love, "Digital reproduction of historical building ornamental components: From 3D scanning to 3D printing," *Automation in Construction*, vol. 76, pp. 85–96, 2017.
- [8] W. Liu, X. Zhou, X. Zhang, and Q. Niu, "Three-dimensional (3D) CAD model lightweight scheme for large-scale assembly and simulation," *International Journal of Computer Integrated Manufacturing*, vol. 28, no. 5, pp. 520–533, 2015.
- [9] M. Wenzel, A. Klinger, and C. Meinel, "Tele-board prototyper - Distributed 3D modeling in a web-based real-time collaboration system," in *Proceedings of the 2016 International Conference on Collaboration Technologies and Systems, CTS 2016*, pp. 446–453, usa, November 2016.
- [10] L. Zhang, L. Liu, C. Gotsman, and H. Huang, "Mesh reconstruction by meshless denoising and parameterization," *Computers and Graphics*, vol. 34, no. 3, pp. 198–208, 2010.
- [11] X. Huang, X. Chen, T. Tang, and Z. Huang, "Marching cubes algorithm for fast 3D modeling of human face by incremental data fusion," *Mathematical Problems in Engineering*, vol. 2013, Article ID 203609, 2013.
- [12] M. Centin, N. Pezzotti, and A. Signoroni, "Poisson-driven seamless completion of triangular meshes," *Computer Aided Geometric Design*, vol. 35/36, pp. 42–55, 2015.

- [13] T. Garrett, S. Debernardis, J. Oliver, and R. Radkowski, "Poisson Mesh Reconstruction for Accurate Object Tracking with Low-Fidelity Point Clouds," *Journal of Computing and Information Science in Engineering*, vol. 17, no. 1, Article ID 011003, 2017.
- [14] J.-L. Peyrot, F. Payan, and M. Antonini, "From stereoscopic images to semi-regular meshes," *Signal Processing: Image Communication*, vol. 40, pp. 97–110, 2016.
- [15] Y. Liu, F. Wang, A. M. Dobaie, G. He, and Y. Zhuang, "Comparison of 2D image models in segmentation performance for 3D laser point clouds," *Neurocomputing*, vol. 251, pp. 136–144, 2017.
- [16] L. Li, C. Li, Y. Tang, and Y. Du, "An integrated approach of reverse engineering aided remanufacturing process for worn components," *Robotics and Computer-Integrated Manufacturing*, vol. 48, pp. 39–50, 2017.
- [17] J. Roth, Y. Tong, and X. Liu, "Adaptive 3D Face Reconstruction from Unconstrained Photo Collections," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2127–2141, 2017.
- [18] D. Boltcheva and B. Lévy, "Surface reconstruction by computing restricted Voronoi cells in parallel," *Computer-Aided Design*, vol. 90, pp. 123–134, 2017.
- [19] C. Marion and J. Jomier, "Real-time collaborative scientific WebGL visualization with WebSocket," in *Proceedings of the 17th International Conference on 3D Web Technology, Web3D 2012*, pp. 47–50, USA, August 2012.
- [20] B. Zimmer and A. Kerren, "Harnessing WebGL and WebSockets for a Web-based collaborative graph exploration tool," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9114, pp. 583–598, 2015.
- [21] F. Mwalongo, M. Krone, M. Becher, G. Reina, and T. Ertl, "GPU-based remote visualization of dynamic molecular data on the web," *Graphical Models*, vol. 88, pp. 57–65, 2016.
- [22] L. Renambot, T. Marrinan, J. Aurisano et al., "SAGE2: A collaboration portal for scalable resolution displays," *Future Generation Computer Systems*, vol. 54, pp. 296–305, 2016.
- [23] J. Alnihoud and R. Mansi, "An enhancement of major sorting algorithms," *International Arab Journal of Information Technology*, vol. 7, no. 1, pp. 55–62, 2010.
- [24] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [25] R. Radkowski, "Object Tracking With a Range Camera for Augmented Reality Assembly Assistance," *Journal of Computing and Information Science in Engineering*, vol. 16, no. 1, p. 011004, 2016.
- [26] J. Vince, *Quaternions for Computer Graphics*, Springer, London, UK, 2011.
- [27] J. M. Lee, "Fast k-Nearest Neighbor Searching in Static Objects," *Wireless Personal Communications*, vol. 93, no. 1, pp. 147–160, 2017.
- [28] M. Greenspan and M. Yurick, "Approximate k-d tree search for efficient ICP," in *Proceedings of the 4th International Conference on 3-D Digital Imaging and Modeling (3DIM '03)*, pp. 442–448, IEEE, October 2003.
- [29] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3D point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 698–700, 1987.
- [30] J. H. Lee, S. Y. Hwang, and C. H. Ryu, "Localization between curved shell plate and its unfolded shape in different coordinate systems for ship-hull plate forming," *Mathematical Problems in Engineering*, vol. 2011, Article ID 257804, 2011.
- [31] X. Wang, J. Cai, Z. Wu, and M. Zhou, "Normal Estimation and Normal Orientation for Point Cloud Model Based on Improved Local Surface Fitting," *Journal of Computer-Aided Design Computer Graphics*, vol. 27, no. 4, pp. 614–620, 2015.
- [32] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, pp. 61–70, Cagliari, Sardinia, Italy, 2006.
- [33] W. E. Lorensen and H. E. Cline, "Marching cubes: a high resolution 3D surface construction algorithm," *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.
- [34] Y. Sheng, A. Sourin, G. G. Castro, and H. Ugail, "A PDE method for patchwise approximation of large polygon meshes," *The Visual Computer*, vol. 26, no. 6-8, pp. 975–984, 2010.
- [35] L. Sun, X. Suo, Y. Liu, M. Zhang, and L. Han, "3D Modeling of Transformer Substation Based on Mapping and 2D Images," *Mathematical Problems in Engineering*, vol. 2016, Article ID 9320502, 2016.
- [36] D. Wang and J. Shao, "The HTML5 graphic and image processing technology," *Agro FOOD Industry Hi Tech*, vol. 28, no. 3, pp. 2956–2959, 2017.
- [37] J. Verdú, J. J. Costa, and A. Pajuelo, "Dynamic web worker pool management for highly parallel javascript web applications," *Concurrency Computation*, vol. 28, no. 13, pp. 3525–3539, 2016.
- [38] S. Sun, M. Zhang, and Z. Gou, "Smoothing Algorithm for Planar and Surface Mesh Based on Element Geometric Deformation," *Mathematical Problems in Engineering*, vol. 2015, Article ID 435648, 2015.

## Research Article

# An Improved Fractional-Order Optical Flow Model for Motion Estimation

Bin Zhu <sup>1</sup>, Lianfang Tian,<sup>1</sup> Qiliang Du <sup>1</sup>, Qiuxia Wu <sup>2</sup>, and Lixin Shi <sup>1</sup>

<sup>1</sup>School of Automation Science and Engineering, South China University of Technology, Guangzhou, Guangdong 510640, China

<sup>2</sup>School of Software Engineering, South China University of Technology, Guangzhou, Guangdong 510640, China

Correspondence should be addressed to Qiliang Du; [qldu@scut.edu.cn](mailto:qldu@scut.edu.cn)

Received 21 July 2017; Accepted 22 November 2017; Published 10 May 2018

Academic Editor: Gabriela Marinoschi

Copyright © 2018 Bin Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Horn and Schunck (HS) optical flow model cannot preserve discontinuity of motion estimation and has low accuracy especially for the image sequence, which includes complex texture. To address this problem, an improved fractional-order optical flow model is proposed. In particular, the fractional-order Taylor series expansion is applied in the brightness constraint equation of the HS model. The fractional-order flow field derivative is also used in the smoothing constraint equation. The Euler-Lagrange equation is utilized for the minimization of the energy function of the fractional-order optical flow model. Two-dimensional fractional differential masks are proposed and applied to the calculation of the model simplification. Considering the spatiotemporal memory property of fractional-order, the algorithm preserves the edge discontinuity of the optical flow field while improving the accuracy of the estimation of the dense optical flow field. Experiments on Middlebury datasets demonstrate the predominance of our proposed algorithm.

## 1. Introduction

Since the optical flow was proposed by Gibson in 1950, many scholars have done research on it. In these studies, the HS optical flow model proposed by Horn and Schunck [1] in 1981 achieved high reliability and accuracy and accordingly began to attract more and more scholarly attention. But the accuracy of the motion estimation with the HS optical flow model would be greatly reduced in the image sequence, which would include complex texture and nonrigid motion; meanwhile the discontinuity of motion estimation is difficult to retain.

Many scholars have devoted themselves to the modification or improvement of HS model to increase the accuracy [2], deal with large displacement motion [3], solve the occlusion problem [4], and track nonrigid motion targets [5, 6]. All these models are based on first-order differentiation techniques.

Lv et al. [7] added a nonlinear fourth-order diffusive term to the Euler-Lagrange equations of the variational TV model, and the improved model can preserve sharp jump discontinuities, but it would deteriorate when too much noise exists. A high-order total variation minimization method for

image deblurring and denoising was proposed by Chan et al. [8]; the model can filter out noise while the edge discontinuity was retained. But details of the edge were neglected.

Because of its ability to preserve the texture details of the smooth region while highlighting the image edge feature and its spatiotemporal memory of the target point neighborhood, a fractional differential is applied to many image processing fields [9], such as image denoising [10], image enhancement [11], and motion estimation [12–16]. In [12, 13], the fractional-order smoothing constraint equation was used in the HS optical flow model to preserve the discontinuity of motion estimation, but it does not consider the correlation of the pixel intensity. Although the edge discontinuity of the image can be preserved to a certain extent, the accuracy of the optical flow field estimation is not high. In order to further enhance the accuracy of motion estimation, an improved fractional-order optical flow model is proposed in this paper.

The main characteristics of the present paper can be summarized as follows:

- (1) Fractional-order was first used in a brightness constraint equation of the HS optical flow model.

- (2) A two-dimensional fractional-order differential mask, which was used to compute dual fractional-order differentials, was first proposed and applied to the simplification of the improved fractional-order optical flow model.
- (3) Combined with the Euler-Lagrange equation and two-dimensional fractional-order differential masks, the iteration formula of the optical flow field estimation of the improved fractional-order optical flow model was constructed, which is different from any of the former counterparts.

## 2. The Construction of Two-Dimensional Fractional-Order Differential Masks

The mathematical preliminaries of the fractional-order theory used in this paper can be found in [17, 18].

In this chapter, six two-dimensional fractional differential masks were presented. There are  $W_{xx}^{(\mu)}(k)$ ,  $W_{xy}^{(\mu)}(k, z)$ ,  $W_{xt}^{(\mu)}(k, p)$ ,  $W_{yy}^{(\mu)}(z)$ ,  $W_{yx}^{(\mu)}(k, z)$ ,  $W_{yt}^{(\mu)}(z, p)$ , where  $\mu$  is the order of fractional differentials.

$$W_{xy}^{(\mu)}(z, k) = \begin{bmatrix} \frac{(-\mu)(-\mu+1)}{2} & \frac{(-\mu)(-\mu+1)}{2} & \frac{(-\mu)(-\mu+1)}{2} & \frac{(-\mu)(-\mu+1)}{2} & \frac{(-\mu)(-\mu+1)}{2} \\ \frac{(-\mu)(-\mu+1)}{2} & -\mu & -\mu & -\mu & \frac{(-\mu)(-\mu+1)}{2} \\ \frac{(-\mu)(-\mu+1)}{2} & -\mu & 8 & -\mu & \frac{(-\mu)(-\mu+1)}{2} \\ \frac{(-\mu)(-\mu+1)}{2} & -\mu & -\mu & -\mu & \frac{(-\mu)(-\mu+1)}{2} \\ \frac{(-\mu)(-\mu+1)}{2} & \frac{(-\mu)(-\mu+1)}{2} & \frac{(-\mu)(-\mu+1)}{2} & \frac{(-\mu)(-\mu+1)}{2} & \frac{(-\mu)(-\mu+1)}{2} \end{bmatrix}. \quad (4)$$

Also,

$$D_x^\alpha D_t^\alpha I(i, j, t) = \sum_{p=0}^T \sum_{k=0}^L W_{xt}^{(\mu)}(k, p) I(i-k, j, t) I(i, j, t-p), \quad (5)$$

where  $k, p \in Z, k \leq L, P \leq T$ . When  $T = L = 2$ ,

$$W_{xt}^{(\mu)}(k, p) = \begin{bmatrix} 1 & -\mu & \frac{(-\mu)(-\mu+1)}{2} \\ -\mu & -\mu & \frac{(-\mu)(-\mu+1)}{2} \\ \frac{(-\mu)(-\mu+1)}{2} & \frac{(-\mu)(-\mu+1)}{2} & \frac{(-\mu)(-\mu+1)}{2} \end{bmatrix}. \quad (6)$$

What is more, we can get  $W_{yy}^{(\mu)}(z) = W_{xx}^{T(\mu)}(k)W_{yx}^{(\mu)}(k, z) = W_{xy}^{(\mu)}(k, z)$ ,  $W_{yt}^{(\mu)}(z, p) = W_{xt}^{(\mu)}(k, p)$ .

Normalization should be done in all the mask items. More details about the expansion of dual fractional-order derivatives can be found in [12].

We define  $W_{xx}^{(\mu)}(k)$  as the two-dimensional fractional differential mask of dual derivatives on the  $x$ -axis. The two-dimensional fractional derivatives of order  $\mu$  of a discrete image brightness function  $I(i, j, t)$  on the  $x$ -axis can be defined as

$$D_x^\mu D_x^\mu I(i, j, t) = \sum_{k=-L}^L W_{xx}^{(\mu)}(k) I(i-k, j, t). \quad (1)$$

$L$  is the mask size, when  $L = 2$ :

$$W_{xx}^{(\mu)}(k) = \begin{bmatrix} \frac{(-\mu)(-\mu+1)}{2} & -\mu & -\mu & \frac{(-\mu)(-\mu+1)}{2} \end{bmatrix}. \quad (2)$$

Similarly,

$$D_x^\mu D_y^\mu I(i, j, t) = \sum_{z=-L}^L \sum_{k=-L}^L W_{xy}^{(\mu)}(z, k) I(i+z, j+k, t), \quad (3)$$

where  $k, z \in Z, k \leq L, z \leq L$ , when  $L = 2$ :

## 3. Improved Fractional-Order Optical Flow Model

**3.1. The Fractional-Order Optical Flow Model.** We substitute the brightness constraint equation of the HS model with the fractional-order Taylor series expansion and get the fractional-order brightness constraint equation as

$$\frac{D_x^\alpha I}{\Gamma(\alpha+1)} u^\alpha + \frac{D_y^\alpha I}{\Gamma(\alpha+1)} v^\alpha + \frac{D_t^\alpha I}{\Gamma(\alpha+1)} = 0, \quad (7)$$

where  $I$  is the abbreviation of  $I(i, j, t)$ .

Combined with (2.19) and (2.20) in [12], we can get the fractional-order optical flow model in this paper:

$$E(U) = \int_{\Omega} \left( \left( \frac{D_x^\alpha I}{\Gamma(\alpha+1)} u^\alpha + \frac{D_y^\alpha I}{\Gamma(\alpha+1)} v^\alpha + \frac{D_t^\alpha I}{\Gamma(\alpha+1)} \right)^2 + \lambda (|D^\beta u|^2 + |D^\beta v|^2) \right) dX. \quad (8)$$

3.2. *Numerical Algorithm.* The Euler-Lagrange equation [19] was applied in this paper to minimize energy function  $E(U)$ .

After simplification, we get the equation as follows:

$$\begin{aligned} \frac{D_x^\alpha I}{\Gamma(\alpha+1)} u^{\alpha-1} M + \lambda (D_x^\beta D_x^\beta u + D_y^\beta D_y^\beta u) &= 0 \\ \frac{D_y^\alpha I}{\Gamma(\alpha+1)} v^{\alpha-1} M + \lambda (D_x^\beta D_x^\beta v + D_y^\beta D_y^\beta v) &= 0, \end{aligned} \quad (9)$$

where

$$M = \frac{D_x^\alpha I}{\Gamma(\alpha+1)} u^\alpha + \frac{D_y^\alpha I}{\Gamma(\alpha+1)} v^\alpha + \frac{D_t^\alpha I}{\Gamma(\alpha+1)}. \quad (10)$$

After a series of computations, the iteration formula for the proposed fractional-order optical flow model can finally be written as

$$\begin{aligned} \nabla^\alpha I_{xx} u^{2\alpha-1} + \nabla^\alpha I_{xy} u^{\alpha-1} v^\alpha + \nabla^\alpha I_{xt} u^{\alpha-1} \\ + \lambda \sum_{k=-L}^L \sum_{z=-L}^L W_{xy}^{(\beta)}(k, z) u(i-k, j-z) &= 0 \\ \nabla^\alpha I_{yy} v^{2\alpha-1} + \nabla^\alpha I_{yx} v^{\alpha-1} u^\alpha + \nabla^\alpha I_{yt} v^{\alpha-1} \\ + \lambda \sum_{k=-L}^L \sum_{z=-L}^L W_{yx}^{(\beta)}(k, z) v(i-k, j-z) &= 0, \end{aligned} \quad (11)$$

where  $u(i-k, j-z)$ ,  $v(i-k, j-z)$  represented the optical flow vector in the neighborhood of point  $(i, j)$ , and

$$\begin{aligned} \nabla^\alpha I_{xx}(i, j, t) &= \frac{\sum_{k=-L}^L W_{xx}^{(\alpha)}(k) I(i-k, j, t)}{\Gamma^2(\alpha+1)} \\ \nabla^\alpha I_{xy}(i, j, t) &= \frac{\sum_{k=-L}^L \sum_{z=-L}^L W_{xy}^{(\alpha)}(k, z) I(i-k, j-z, t)}{\Gamma^2(\alpha+1)} \\ \nabla^\alpha I_{xt}(i, j, t) &= \frac{\sum_{p=0}^T \sum_{k=0}^L W_{xt}^{(\alpha)}(k, p) I(i-k, j, t) I(i, j, t-p)}{\Gamma^2(\alpha+1)} \\ \nabla^\alpha I_{yy}(i, j, t) &= \frac{\sum_{z=-L}^L W_{yy}^{(\alpha)}(k) I(i, j-z, t)}{\Gamma^2(\alpha+1)} \\ \nabla^\alpha I_{yx}(i, j, t) &= \frac{\sum_{k=-L}^L \sum_{z=-L}^L W_{yx}^{(\alpha)}(z, k) I(i-k, j-z, t)}{\Gamma^2(\alpha+1)} \\ \nabla^\alpha I_{yt}(i, j, t) &= \frac{\sum_{p=0}^T \sum_{z=0}^L W_{yt}^{(\alpha)}(z, k) I(i, j-z, t) I(i, j, t-p)}{\Gamma^2(\alpha+1)}. \end{aligned} \quad (12)$$

After repeated iteration utilizing the Gauss-Seidel iteration method, the accurate optical flow field can be achieved, and the convergence of the algorithm can be demonstrated by the method mentioned in [13].

## 4. Experimental Results and Analysis

We have conducted our experiments from the following three aspects: (1) the comparison of the performance of previously tested optical flow algorithms; (2) the influence of the order of fractional-order differentials for the algorithm; and (3) the influence of the size of the dual fractional differential mask window for the algorithm. The Middlebury datasets [20] are chosen for evaluating our proposed algorithm for its characteristics of having different structures and attributes, like textures, blurred images, discontinuities in motion, shadows, nonrigid motion, and so on.

The angular error (AE) and flow endpoint error (EE) are selected for evaluation. As we are concerned with the performance of the algorithm in the whole image, the AE/EE in the entire image (AVAE/AVEE) are used to visually evaluate our proposed optical model.

For convenient description, we call the HS optical flow model the HS model, the model in [12] the FOVOF model, and our proposed model the DFOVOF model. The mask window size is  $L$ .

Figure 1 shows the optical flow field of the HS model, the FOVOF model, and the DFOVOF model in different image sequences. In the DFOVOF model,  $\alpha = 0.8$ ,  $\beta = 1.4$ ,  $L = 5$ . Among them, different colors represent different movement directions and different color depth represents different speed of movement. The deeper the color, the higher the speed.

The first row in Figure 1 shows the acquired optical flow field produced by the traditional HS model. It can be seen that the edge of the image discontinuity is not well reserved and the details of the image texture are neglected.

The second row shows the optical flow field obtained by the FOVOF model. The edge discontinuity is reserved to some extent, but the contour of the object is not very clear.

In the third row, the optical flow field obtained by the DFOVOF optical flow model is presented. The contour edge is much more obvious; meanwhile the details of image texture can be seen clearly. The DFOVOF model expands the brightness constraint equation in the form of a Taylor fractional-order, and the local brightness correlation of the image is also added for the calculation of the optical flow field, which further reduces the error and improves the accuracy of the model.

In order to evaluate the influence of fractional-order parameters on the performance of the model and find out the correlation between the fractional-order and the image characteristics, the AVAE/AVEE of the optical flow field of each image sequence at different fractional orders is estimated, as shown in Figure 2. At this time we choose  $L = 2$ . The smaller the value of AVAE/AVEE, the higher the accuracy of the optical flow field estimation. The first row is the AVAE and AVEE of image sequence "Yosemite"; the second row is the AVAE and AVEE of image sequence "Army." The third row is the AVAE and AVEE of image sequence "Urban."

We can find in Figure 2 that the minimum AVAE/AVEE  $\alpha$  is distributed in a range of (0.3, 0.8) and the minimum AVAE/AVEE  $\beta$  is in a range of (1.3, 1.4); the simpler the image texture, the smoother the surface representing the variation of the value of AVEE/AVAE to variable  $\alpha$  and  $\beta$ ;

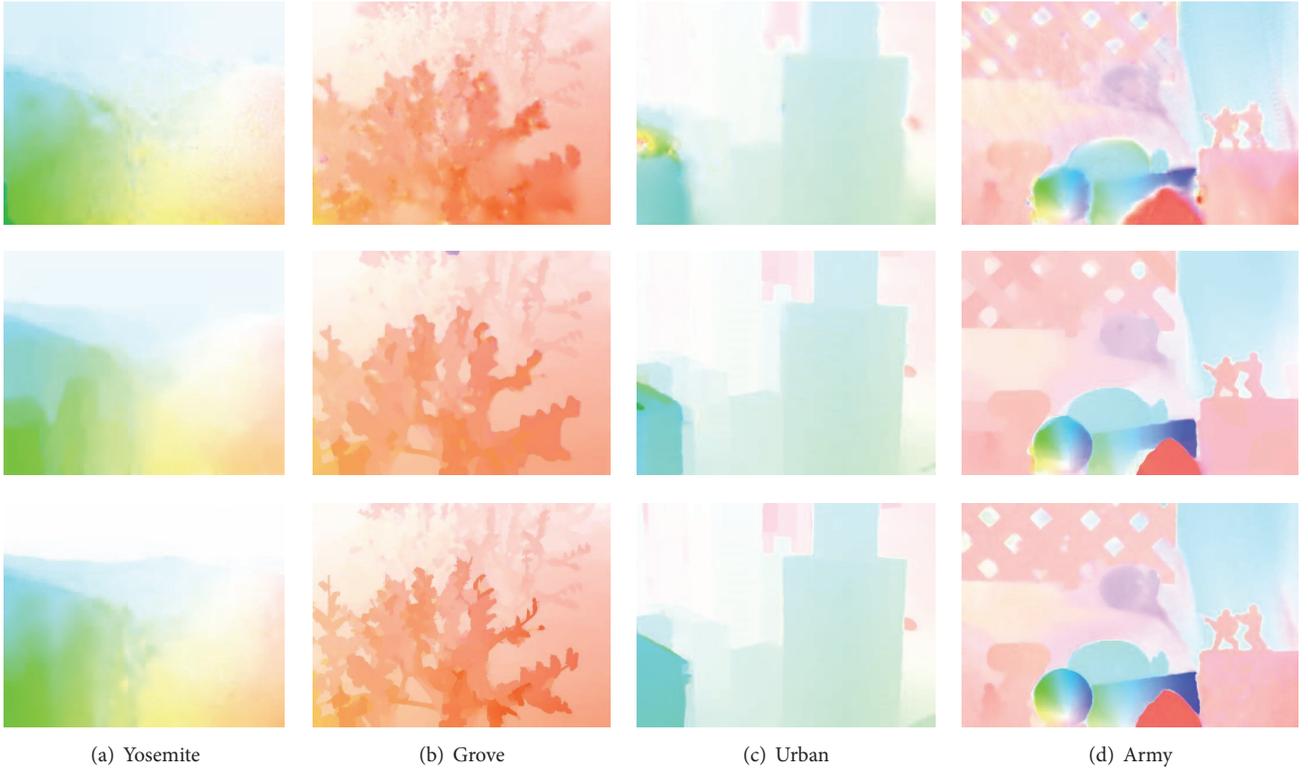


FIGURE 1: Optical flow results of the estimated optical flows by color maps. The first row shows the results of the HS model, the second row shows the results of the HS model, and the third shows the results of the DFOVOF model.

the more complex the image texture, the steeper the surface; the surface would also increase in incline when there exists a nonrigid motion in image sequences. Thus, it is clear that the algorithm presented here can improve the accuracy of motion estimation, especially for image sequences with complex texture and nonrigid motion. In conclusion, we choose a small  $\alpha$  for a complex texture image and choose a relatively large  $\alpha$  for a simple texture image sequence. What is more, nonrigid motion would also make the optimal  $\alpha$  small. The optimal  $\beta$  changes little in different image sequence.

To test the effect of the mask window size  $L$  on the algorithm's performance and find out the correlation between mask window length  $L$  and image features, we calculate the AVEE of optical flow estimation of six image sequences at different mask window size  $L$  which was shown in Table 1, where  $\alpha = 0.8$ ,  $\beta = 1.3$ , where (a) is Venus; (b) is dimetrodon; (c) is hydrangea; (d) is Rubberwhale; (e) is grove; and (f) is urban. For comparison, the AVEE of H-S model and FOVOF model were also added to the table. For comparison, the AVEE values for the optical flow field estimation of the H-S model and the FOVOF model are also added to the table.

We can see in Table 1 that the value of AVEE reduces further when applied to the DFOVOF model; in the DFOVOF model the error rate becomes smaller when the mask window size  $L$  increases—in general, we get the smallest AVEE value when  $L$  is within (2, 7). For all the images, AVEE will decrease first when  $L$  rises; after AVEE arrives at its optimal value, it will then increase when  $L$  rises. The more complex the image

TABLE 1: The AVEE of six image sequences at different mask window size.

Model	Image					
	(a)	(b)	(c)	(d)	(e)	(f)
H-S	0.5494	0.1908	0.2967	0.2268	0.2919	1.3944
FOVOF	0.5334	0.1861	0.2885	0.2071	0.2910	1.1085
DFOVOF						
$L = 2$	0.5220	0.1848	0.2860	0.2046	0.2874	1.0646
$L = 3$	0.5216	0.1843	0.2854	0.2043	0.2872	1.0603
$L = 4$	0.5204	0.1838	0.2845	0.2024	0.2866	1.0534
$L = 5$	0.5193	0.1826	0.2837	0.1993	0.2860	1.0546
$L = 6$	0.5175	0.1815	0.2825	0.2026	0.2852	1.0573
$L = 7$	0.5183	0.1823	0.2806	0.2034	0.2856	1.0592
$L = 8$	0.5197	0.1831	0.2810	0.2043	0.2863	1.0623

texture is or the more nonrigid the motion is, the smaller the value of  $L$  will be.

In conclusion, we should choose a large value  $L$  for relatively simple image sequences and choose a small value  $L$  for complex texture images and nonrigid motion images.

## 5. Conclusion

We have presented an improved fractional-order optical flow model, which applies fractional-order both in brightness

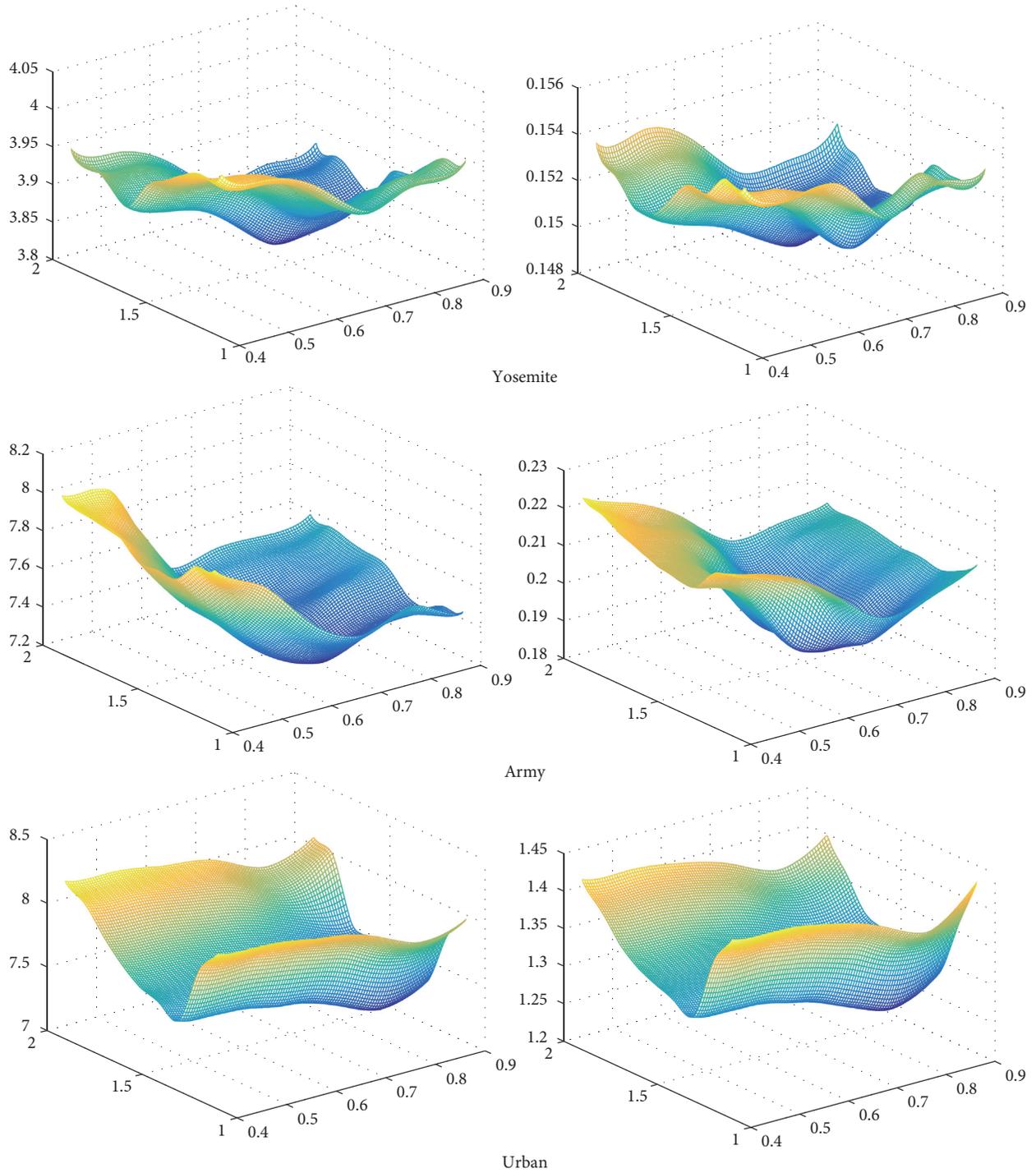


FIGURE 2: The AVAE and AVEE of the optical flow field of three different image sequences at different fractional-order.

constraint equation and in smoothness constraint equation of the HS model. What is more, we have proposed two-dimensional fractional differential masks and used them to simplify the computation of dual fractional-order derivatives. Experiments show that the model created here improves accuracy further. We have also found the relationships between the values of  $\alpha$ ,  $\beta$ , and  $L$  and the image content. So different image content should use different  $\alpha$ ,  $\beta$ , and  $L$ ,

but the algorithm we employ utilizes the same  $\alpha$ ,  $\beta$ , and  $L$  in the whole image. Further studies will focus on the adaptive adjustments of the value of  $\alpha$ ,  $\beta$ , and  $L$  in relation to the image content.

**Conflicts of Interest**

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (61503141), Guangdong Province Frontier and Key Technological Innovation Special Funds (2016B090912001), the Production Research Project in Guangzhou (201604010114), the International Cooperation Projects of Science and Technology Information Bureau in Guangzhou City (2012J5100001), Sea Defense Public Welfare Project (201505002), and the Fundamental Research Funds for the Central Universities.

## References

- [1] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1–3, pp. 185–203, 1981.
- [2] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1744–1757, 2012.
- [3] Z. Chen, H. Jin, Z. Lin, S. Cohen, and Y. Wu, "Large displacement optical flow from nearest neighbor fields," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013*, pp. 2443–2450, USA, June 2013.
- [4] X. Zhang, H.-M. Hu, F. Jiang, and B. Li, "Pedestrian detection based on hierarchical co-occurrence model for occlusion handling," *Neurocomputing*, vol. 168, pp. 861–870, 2015.
- [5] W. Li, D. Cosker, Z. Lv, and M. Brown, "Nonrigid optical flow ground truth for real-world scenes with time-varying shading effects," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 231–238, 2017.
- [6] W. Li, D. Cosker, and M. Brown, "Drift robust non-rigid optical flow enhancement for long sequences," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 31, no. 5, pp. 2583–2595, 2016.
- [7] X.-G. Lv, Y.-Z. Song, S.-X. Wang, and J. Le, "Image restoration with a high-order total variation minimization method," *Applied Mathematical Modelling*, vol. 37, no. 16–17, pp. 8210–8224, 2013.
- [8] T. Chan, A. Marquina, and P. Mulet, "High-order total variation-based image restoration," *Society for Industrial and Applied Mathematics*, vol. 22, no. 2, pp. 503–516, 2000.
- [9] Q. Yang, D. Chen, T. Zhao, and Y. Chen, "Fractional calculus in image processing: a review," *Fractional Calculus and Applied Analysis*, vol. 19, no. 5, pp. 1222–1249, 2016.
- [10] D. Chen, Y. Chen, and D. Xue, "Fractional-order total variation image denoising based on proximity algorithm," *Applied Mathematics and Computation*, vol. 257, pp. 537–545, 2015.
- [11] Y.-F. Pu, J.-L. Zhou, and X. Yuan, "Fractional differential mask: a fractional differential-based approach for multiscale texture enhancement," *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 491–511, 2010.
- [12] D. Chen, H. Sheng, Y. Q. Chen, and D. Xue, "Fractional-order variational optical flow model for motion estimation," *Philosophical Transactions of the Royal Society A: Mathematical, Physical & Engineering Sciences*, vol. 371, no. 1990, Article ID 20120148, 2013.
- [13] P. Kumar, S. Kumar, and B. Raman, "A fractional order variational model for the robust estimation of optical flow from image sequences," *Optik - International Journal for Light and Electron Optics*, vol. 127, no. 20, pp. 8710–8727, 2016.
- [14] P. Kumar, S. Kumar, and B. Raman, "A fractional order variational model for tracking the motion of objects in the applications of video surveillance," in *Proceedings of the 8th International Conference on Advanced Computational Intelligence, ICACI 2016*, pp. 117–123, Thailand, February 2016.
- [15] P. Kumar and S. Kumar, "A modified variational functional for estimating dense and discontinuity preserving optical flow in various spectrum," *AEÜ - International Journal of Electronics and Communications*, 2015.
- [16] S. G. Bardeji, I. N. Figueiredo, and E. I. Sousa, "Optical flow with fractional order regularization: variational model and solution method," *Applied Numerical Mathematics*, vol. 114, pp. 188–200, 2017.
- [17] K. S. Miller, "Derivatives of noninteger order," *Mathematics Magazine*, vol. 68, no. 3, pp. 183–192, 1995.
- [18] L. Podlubny, *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution And Some of Their Applications*, Academic Press, 1999.
- [19] O. P. Agrawal, "Formulation of euler-lagrange equations for fractional variational problems," *Journal of Mathematical Analysis and Applications*, vol. 272, no. 1, pp. 368–379, 2002.
- [20] S. Baker, D. Scharstein, J. Lewis, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, March 2011.

## Research Article

# Fast Video Dehazing Using Per-Pixel Minimum Adjustment

Zhong Luan <sup>1,2</sup>, Hao Zeng,<sup>1</sup> Yuanyuan Shang <sup>1,2,3</sup>, Zhuhong Shao,<sup>2,3</sup> and Hui Ding<sup>2,3</sup>

<sup>1</sup>College of Information Engineering, Capital Normal University, Beijing, China

<sup>2</sup>Beijing Advanced Innovation Center for Imaging Technology, Beijing, China

<sup>3</sup>Beijing Engineering Research Center of High Reliable Embedded System, Beijing, China

Correspondence should be addressed to Yuanyuan Shang; [syy@bao.ac.cn](mailto:syy@bao.ac.cn)

Received 21 November 2017; Revised 5 January 2018; Accepted 11 January 2018; Published 12 February 2018

Academic Editor: Ionuț Munteanu

Copyright © 2018 Zhong Luan 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 reduce the computational complexity and maintain the effect of video dehazing, a fast and accurate video dehazing method is presented. The preliminary transmission map is estimated by the minimum channel of each pixel. An adjustment parameter is designed to fix the transmission map to reduce color distortion in the sky area. We propose a new quad-tree method to estimate the atmospheric light. In video dehazing stage, we keep the atmospheric light unchanged in the same scene by a simple but efficient parameter, which describes the similarity of the interframe image content. By using this method, unexpected flickers are effectively eliminated. Experiments results show that the proposed algorithm greatly improved the efficiency of video dehazing and avoided halos and block effect.

## 1. Introduction

In the communities of satellite remote sensing, aviation, shipping, and land transportation, the images and videos obtained by electronic equipment are expected to be clear enough. But, in the real scenes, the existence of haze greatly degrades the quality of the captured images and videos. This not only affects the reliability of the monitoring devices but also may cause potential danger. Therefore, it is imperative to develop a simple and efficient real-time dehazing algorithm.

Despite being an ill-posed problem, single-image dehazing has different types of prior approaches. A priori based methods tend to learn the statistical law of haze-free images. He et al. [1] propose a dark channel prior for haze-free images, and the dehazing effect of this algorithm is impressive. But its performance on the sky region and white objects is unsatisfactory, and the time complexity of soft matting is very high. As the best algorithm at that time, many algorithms improve the DCP method in some particular aspects [2–4]. Chen et al. [5] divide the image into foreground and background based on Fisher's linear discriminant, to process images with a dramatic depth change. Chen et al. [6–9] combine improved DCP method with white balance and local contrast enhances into a unified solution, which achieves good results on

sandstorm weather. Luan et al. [10] maximize image contrast by a cost function that contains the contrast term and the information loss term. This method has obvious advantages in speed; however, some overenhancement appears in their results. Li and Zheng [11] present a simple but effective change of details prior to remove haze from a single image. But this method is mainly stable to local image regions containing objects at different depths. This method works well for most images, but the enhancement for heavy haze regions is not sufficient.

Machine learning has been developing rapidly in recent years. The research focus of dehazing is gradually transformed from image prior to the learning approaches [12–16]. While the effect of dehazing has been substantially improved, undesired results still exist in many challenging scenes, such as street views, thick fog areas, and sky regions. On the other hand, the efficiency of such methods is not dominant.

The main objective of this paper is to develop a fast dehazing algorithm for image and video, where per-pixel method and quad-tree algorithm are utilized. To estimate the transmission map and amend the transmission values that are not accurate enough, we employ the per-pixel method, which can effectively solve the problems of halos and block artifacts near the depth edges and color distortion in the

sky area, also greatly enhancing the speed of estimating the transmission. To further improve the efficiency, the improved quad-tree algorithm is adopted to estimate the atmospheric light. Meanwhile, the proposed algorithm is applied to video dehazing. By keeping the atmospheric light unchanged in the same scene, the proposed algorithm eliminates unexpected flickers in the video dehazing effectively and it achieves fast speed because of using the correlation between two neighbor frames.

## 2. Single-Image Dehazing

**2.1. Haze Modeling.** Dehazing is a problem of image restoration; the degradation of a haze image is due to the suspended particles in the turbid air. In this paper, the atmospheric scattering model is used to describe the formation process of haze images. Haze removal adopts the atmospheric scattering model that is used widely in the field of computer vision and computer graphics. Mathematically, the atmospheric scattering model [18] is given as

$$I(x) = t(x)J(x) + (1 - t(x))A, \quad (1)$$

where  $I(x)$  is the observed haze image at a pixel position  $x$ ,  $J$  is the original scene radiance,  $A$  is the atmospheric light that represents the intensity of the sky light or background light, and  $t$  is the medium transmission, determined by the distance from the object to the camera and the turbidity of medium. According to this model, the task of dehazing algorithm is to estimate  $t$  and  $A$  from the haze image  $I$ . The accuracy of estimating these two parameters is the key to improve the dehazing process.

**2.2. Atmospheric Light Estimation.** Although the use of global atmospheric light [10] will have a better effect, its speed cannot meet the real-time requirements. In this article, an improved quad-tree algorithm is proposed to estimate the atmospheric light. It should be mentioned that our quad-tree method and the algorithm in [17] have similar results, but our method has faster speed.

The process of our quad-tree method is summarized as follows. The channel minimum map of image is firstly computed, which aims to avoid mistake estimation of the atmospheric light when there are extreme values in some color channels in a local patch. Secondly, the channel minimum map is equally divided into four blocks and then the mean gray values are calculated. Then divide the block whose mean gray value is maximal into four blocks equally. Repeat the above process until the block's size is less than the given threshold. Choose the maximum value of the last block's pixels in the input image  $I$  as the estimation value of atmospheric light. This algorithm converges very fast, and its time complexity is low. The process of the quad-tree algorithm is as shown in Figure 1.

**2.3. Transmission Estimation.** Based on the statistical characteristics on haze-free images, He et al. [1] proposed an empirical regularity that is called dark channel prior. Then they estimated transmission  $t$  as the minimum value in the

local area of the minimum channel. However, there will be obvious halos and block artifacts after haze removal by using the above method; using soft matting or guided filter to refine the transmission map can improve this phenomenon but its time complexity is very high. In the algorithm of this paper, we estimate the transmission map by a very simple method and then try to optimize it. Considering that the minimal channel of the haze image part is larger than the clear part, we estimate the preliminary  $t$  as follows:

$$t(x) = 1 - \varepsilon \min_{c \in \{r, g, b\}} \frac{I^c(y)}{A^c}. \quad (2)$$

Usually, 0.85 is thought to be a proper value for  $\varepsilon$ . And, in accordance with specific information of the actual haze image, the value of  $\varepsilon$  could be adjusted suitably. Equation (2) uses per-pixel method to estimate the transmission, instead of dividing the image into blocks, which can preserve full and precise image details. So there is no need to refine the transmission by soft matting [19] or guided filter [20].

Theoretically, the transmission in a local area with the same scene depth should be uniform. But (2) cancels the minimum filter when estimating the transmission, which leads most details of the input image to be kept in the transmission map. To the pixels whose channel minimum values are relatively large, their transmission values calculated by (2) are smaller than the others in a local patch. So that those pixels will seem to be dim after haze removal.

So the transmission values of the pixels whose channel minimum values are relatively large should be increased properly. Here two parameters  $\Delta$  and  $K$  are introduced:

$$\Delta(x) = \left| \min_{c \in \{r, g, b\}} I^c(x) - \min_{c \in \{r, g, b\}} A^c \right|. \quad (3)$$

$K$  is a given threshold value. The pixels satisfying  $\Delta(x) < K$  are thought to be the pixels whose channel minimum values are relatively large, which increase their transmission values properly. Otherwise, these pixels are thought to be the pixels whose channel minimum values are relatively small, which decrease their transmission values properly in order to promote the contrast. So here an adjustment parameter  $\alpha$  is introduced, and (2) is redefined as

$$t(x) = \alpha(x) \left( 1 - \varepsilon \min_c \frac{I^c(y)}{A^c} \right). \quad (4)$$

$\alpha$  is defined by

$$\alpha(x) = \max \left( \sqrt{\frac{K}{\Delta(x)}}, \mu \right), \quad \mu \in (0, 1]. \quad (5)$$

The parameter  $\mu$  is used to decrease the transmission values properly where channel minimum values are relatively small. The value of  $\mu$  should be greater than 0.7 in case of some pixels becoming too dark. The significance of the square root in (5) is to weaken excessive enhancement for the pixels whose values are very close to the atmospheric light, as shown in Figure 2. Moreover, after haze removal, the output image may be dark such that a light increment  $\Delta l$  should be added to the output image. Figure 3 shows the results of the period before and after optimizing.

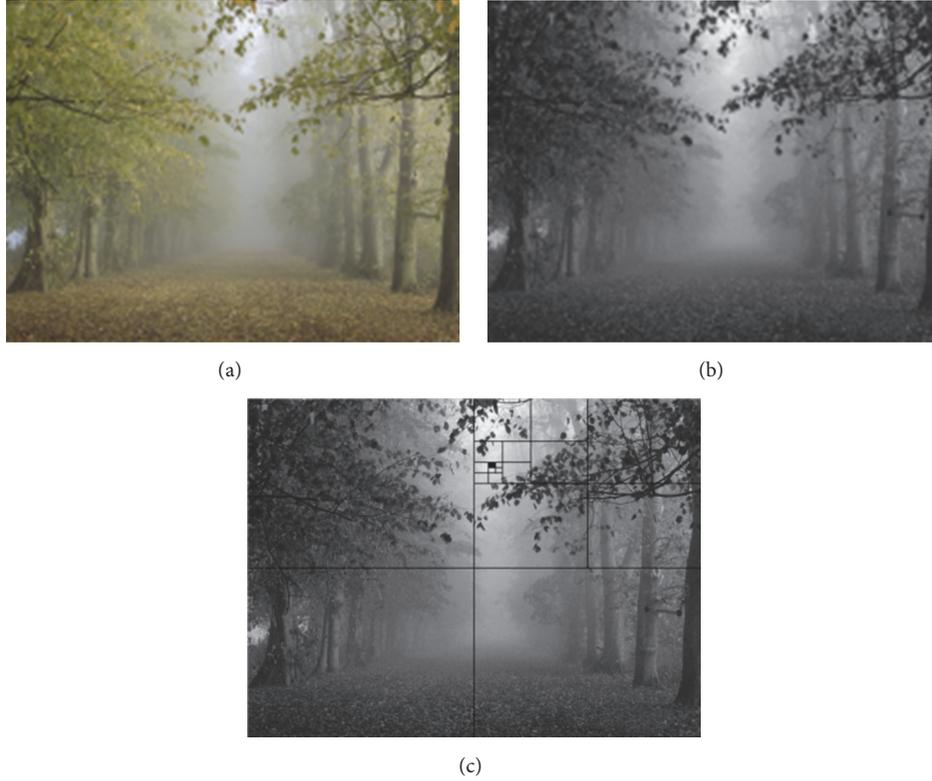


FIGURE 1: Illustration of quad-tree algorithm. (a) Original image. (b) Channel minimum map. (c) Result of quad-tree division.

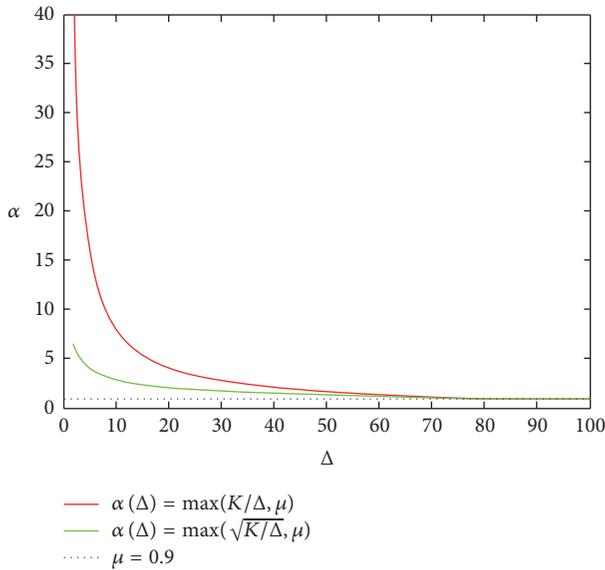


FIGURE 2: Curve of  $\alpha$  relative to  $\Delta$  ( $K = 80$ ).

### 3. Video Haze Removal

**3.1. Eliminating Unexpected Flickers.** In this subsection, the proposed algorithm in Section 2 is applied to video dehazing. When the scene changes in a video, there will be normal flickers. However, in the same scene, smooth transitions

should appear between the adjacent frames. But if we use our method to process a video frame by frame directly without other extra operations, there will be some obvious unexpected flickers even in the same scene. Here we call the factors, which cause this problem, instable factors. From (1) we can see that  $I$ ,  $t$ , and  $A$  may be the possible factors causing this problem. Generally speaking, the original video usually is smooth and does not have unexpected flickers' show. On the other hand, every pixel in a frame is processed independently in our method that it will not be affected by its neighbor pixels, which means  $I$  is a stable factor and we can ignore the influence of  $I$ . From (2) and (4), we can see that  $t(x)$  is determined by  $I(x)$  and  $A$  when  $\epsilon$ ,  $K$ ,  $\mu$ , and  $\Delta I$  are constants. As we know  $I$  is a stable factor, the instability of  $t$  is caused by  $A$ . Based on the above analysis,  $A$  is thought to be the only unstable factor that leads to the unexpected flickers. Figure 5 shows the unexpected flickers between serial frames in the same scene. The values of the estimated atmospheric light of the four frames in Figure 4 are (146, 142, and 141); (159, 142, and 142); (159, 142, and 142); and (145, 141, 140). The difference between the values of the estimated atmospheric light will be magnified by dividing by  $t$  when using (1) to recover the frames.

Actually, the atmospheric light usually keeps unchanged in the same scene in a short period of time. When different atmospheric light values are used to process two neighboring frames in the same scene, it may bring an unexpected flicker. A better strategy is to use the last frame's atmospheric light

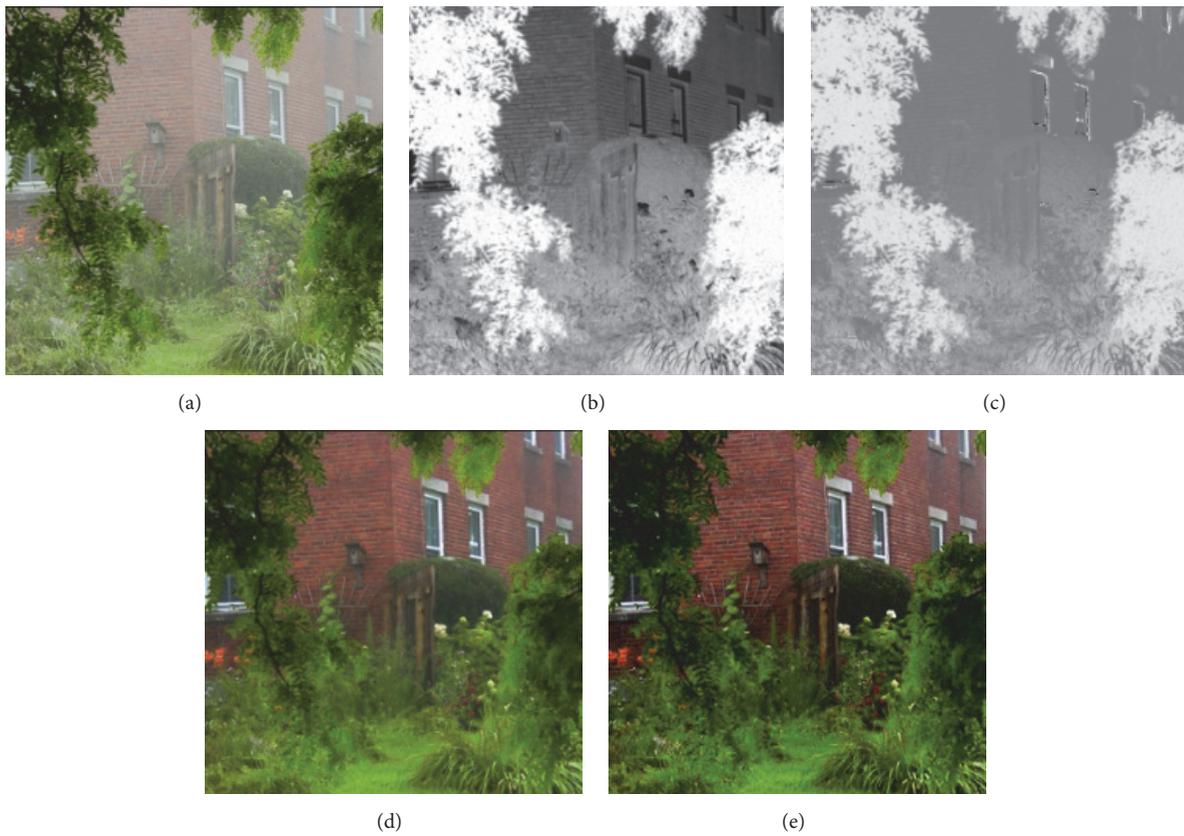


FIGURE 3: Results of before and after optimizing. (a) Input haze image. (b) Estimated transmission map before optimizing. (c) Estimated transmission map after optimizing. (d) Recovered image by (b). (e) Recovered image by (c). ( $\varepsilon = 0.85, K = 80, \mu = 0.9$ , and  $\Delta l = 0$ ).

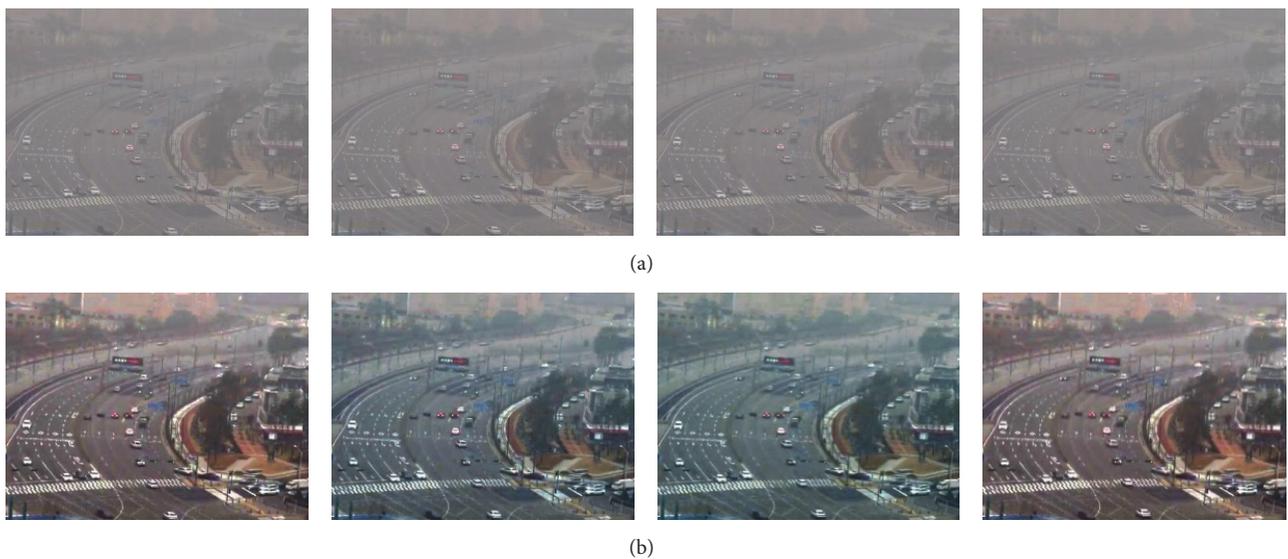


FIGURE 4: Unexpected twinkle in the same scene. (a) Serial input frames. (b) Output frames. ( $\varepsilon = 0.95, K = 35, \mu = 0.9$ , and  $\Delta l = 15$ ).



FIGURE 5: Three short fragments of a video. (a), (b), and (c) are three fragments of continuous frames. The  $\Delta I_{ave}$  values of these frames are shown in Table 1.

value to recover the current frame instead of recalculating. This operation not only can avoid the unexpected flickers but also saves time.

To distinguish whether two neighboring frames are in the same scene, we put forward a simple and efficient method as follows. Firstly, two parameters  $\Delta I$  and  $\Delta I_{ave}$  are, respectively, defined by

$$\begin{aligned} \Delta I(x) &= \text{sum} |I_{\text{current}}(x) - I_{\text{last}}(x)|, \\ \Delta I_{ave} &= \frac{\text{sum}(\Delta I)}{\text{size}(\Delta I)}, \end{aligned} \quad (6)$$

where  $I_{\text{current}}$  and  $I_{\text{last}}$  denote the current input frame and the last input frame. Then, if  $\Delta I_{ave}$  is small enough, the current frame and the last frame are considered in the same scene and the atmospheric light  $A$  is unchanged. Otherwise, we recalculate the atmospheric light  $A$ .

An appropriate threshold is needed to be chosen for  $\Delta I_{ave}$  to distinguish different scene correctly. Figure 5 shows three short fragments of a video. We can see each fragment changes scene at the third frame. Table 1 shows the  $\Delta I_{ave}$  values of frames in Figure 5. It can be observed that, between two neighboring frames, the value of  $\Delta I_{ave}$  is either very small or very large. In this paper, based on a large number of experiments, the threshold of  $\Delta I_{ave}$  is set to 30.

**3.2. Improving Efficiency.** To further enhance the efficiency of video processing, we use the relativity between two neighboring frames. As mentioned above, our method adopts the approach of per-pixel processing to evaluate the transmission, and  $t(x)$  is determined by  $I(x)$  and  $A$  when  $\varepsilon$ ,  $K$ ,  $\mu$ , and

TABLE 1: The  $\Delta I_{ave}$  values of frames in Figure 5.

	Frame 1	Frame 2	Frame 3	Frame 4
Fragment 1	6.24	13.14	120.22	12.10
Fragment 2	5.06	6.00	114.81	7.67
Fragment 3	21.74	13.10	209.23	3.87

$\Delta I$  are constants. However, from (1) we can see that  $J(x)$  is determined by  $I(x)$ ,  $A$ , and  $t(x)$ . So, in our method,  $J(x)$  is only determined by  $I(x)$  and  $A$ .

In Section 3.1, we keep the atmospheric light  $A$  of the current frame the same with the last frame if they are in the same scene. When the current frame and the last frame are in the same scene, and  $I_{\text{current}}(x)$  is equal to  $I_{\text{last}}(x)$ ,  $J_{\text{current}}(x)$  must be equal to  $J_{\text{last}}(x)$  where  $J_{\text{current}}$  is the current output frame and  $J_{\text{last}}$  is the last output frame. In this case, we do not need to recalculate  $J_{\text{current}}(x)$ . For the neighboring frames in the same static scene, there are only a few pixels needed to be recalculated for haze removal. This will save a lot of time for us. But when the frames are in a dynamic scene, almost all the pixel values will change, such that almost all these pixel values of the current frame need to be recalculated. We can judge whether  $I_{\text{current}}(x)$  is equal to  $I_{\text{last}}(x)$  or not by judging whether  $\Delta I(x)$  is equal to zero or not.

## 4. Experimental Results

To evaluate the performance of the proposed algorithm, experiments are performed on a computer with Intel(R) Core(TM) i7-6800K CPU @3.40 GHz and 32.00 GB RAM.

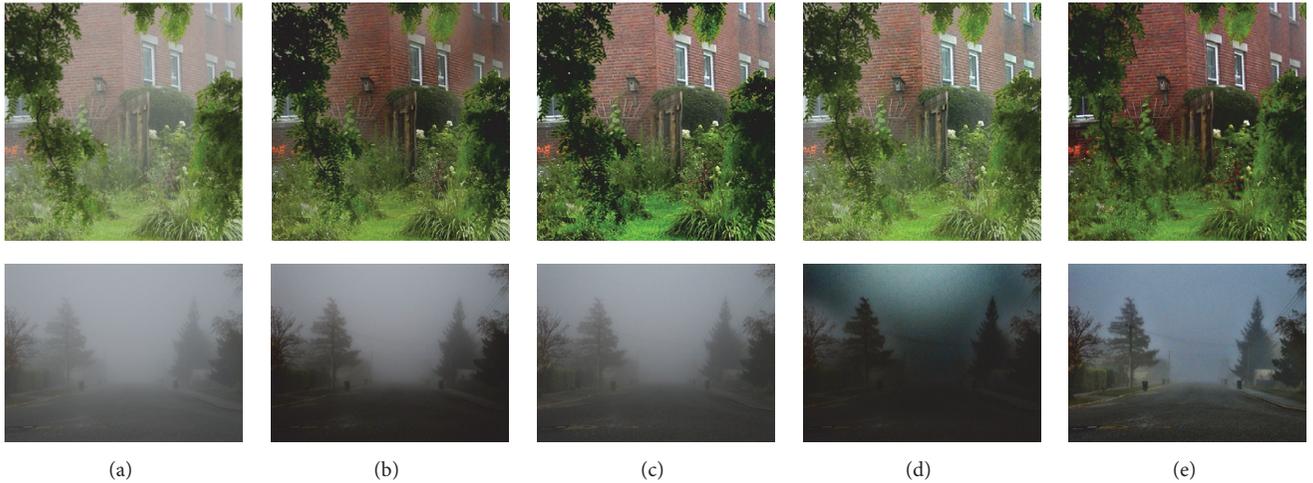


FIGURE 6: Results of image dehazing by different algorithms: (a) original image. (b) Cai et al.'s result [13]. (c) Li and Zheng's result [11]. (d) Kim et al.'s result [17]. (e) Our result ( $\epsilon = 0.85, K = 80, \mu = 0.9, \Delta l = 0$  and  $\epsilon = 0.85, K = 60, \mu = 0.8, \text{ and } \Delta l = 0$ ).

TABLE 2: Time comparison of different algorithms.

Image resolution	Kim et al. [17]	Cai et al. [13]	Li and Zheng [11]	Ours
$510 \times 510$	0.28	2.08	0.82	0.19
$576 \times 768$	0.48	3.43	1.49	0.23
$1,024 \times 768$	0.94	5.91	2.58	0.65
$2,048 \times 1,280$	2.91	19.14	7.69	1.96

For the image haze removal, the experiments are implemented by using MATLAB R2017a and, for the video haze removal, the OpenCV is used.

**4.1. Single-Image Haze Removal.** Experiments have been done firstly to evaluate the efficiency of the proposed scheme for single image, and its performance is compared with those of the work reported in [11, 13, 17].

The dehazing results by using different algorithms are presented in Figure 6. In the “brick wall” photo, the plant at the foreground undergoes oversaturation in (b) and (c). At the same time, information loss occurred in (b) due to the overdarkness of some areas. There are obviously halo and block artifacts near the depth edges in (e). Our method generates better result (e) than other approaches; (e) not only can reserve almost all the details information of depth edge but also avoids generating halo and block artifacts. In the “road” photo, haze is not removed well in (b) and (c). Noise is amplified in (e) which was overenhanced. The contrast of (e) is significantly better than the other results.

Table 2 shows the time comparison of different algorithms performed on varying image size. It can be observed that our proposed algorithm cost the least time whatever the images size is. The result demonstrates that the proposed algorithm is practical to image dehazing.

TABLE 3: Speed comparison between Kim et al.'s algorithm and our algorithm.

	Video 1	Video 2	Video 3	Video 4
Frame size	$640 \times 480$	$480 \times 360$	$640 \times 356$	$640 \times 480$
Kim et al.'s results	56.1 fps	86.0 fps	80.6 fps	56.8 fps
Our results	98.5 fps	175.1 fps	113.3 fps	95.0 fps

**4.2. Video Haze Removal.** Next, experiments are carried out to see the validity of the proposed algorithm when it is applied to video dehazing. It should be noted that, in the video part, we only compare our algorithm to [17] for three reasons: (1) the efficiency of algorithm [11, 13] is not enough for real-time video dehazing, which is the main purpose of this article. (2) In the parameter optimization step, both [11] and [13] use the guided filter [20] to smooth their parameter maps, which would cause some halo artifacts as in [17]. (3) Without the atmospheric light strategy we proposed in the video dehazing part, none of [11, 13, 17] can eliminate unexpected flickers, which leads to approximate experimental results.

The results are shown in Figures 7 and 8. It can be seen that Kim's algorithm not only generated obvious halos and block artifacts near the depth edges but also brought distinct local flickers to the video. The problem of color distortion still exists. However, our algorithm can overcome the above-mentioned shortcomings.

Table 3 lists the cost time of Kim's algorithm and ours tested on different sizes. Obviously, our proposed algorithm is more efficient. It attributes to the fast image haze removal and the timesaving operation.

## 5. Conclusion

This paper introduces a real-time dehazing algorithm for single image and video, the transmission parameter  $t$  is



FIGURE 7: Results of video dehazing by different algorithms: (a) serial input frames. (b) Kim et al.'s result. (c) Our result ( $\epsilon = 0.95, K = 35, \mu = 0.9$ , and  $\Delta l = 15$ ).

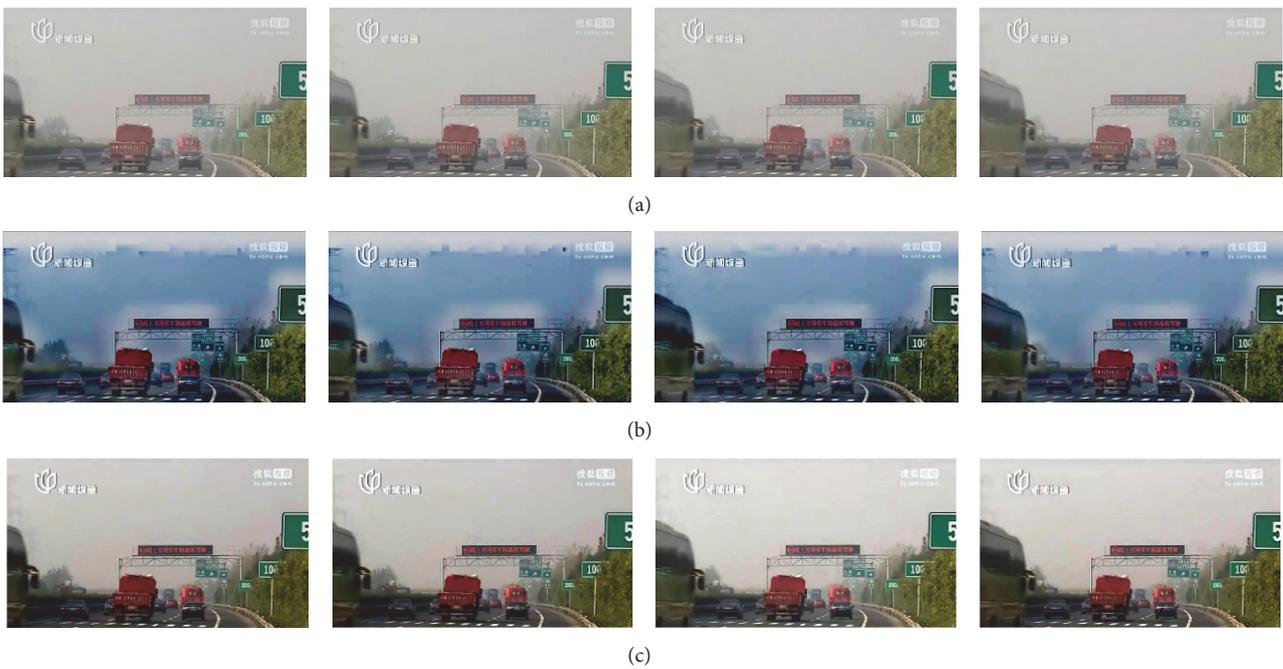


FIGURE 8: Results of video dehazing by different algorithms: (a) serial input frames. (b) Kim et al.'s result. (c) Our result ( $\epsilon = 0.85, K = 90, \mu = 0.80$ , and  $\Delta l = 20$ ).

estimated by a per-pixel method instead of a block method, and the airlight  $A$  is estimated by our improved quad-tree method. Experiment results demonstrated the superior efficiency of the proposed algorithm. Future work will concentrate on applying the proposed algorithm to the images with deep scene depth and figure out the regulation of how to adjust the parameters in our algorithm when the scene changes in a video.

### Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

### Acknowledgments

This work was supported by the Project of Beijing Excellent Talents (2016000020124G088), Beijing Municipal Education Research Plan Project (SQKM201810028018), Support Project of High-Level Teachers in Beijing Municipal Universities in the Period of 13th Five-Year Plan, Natural Science Foundation of Beijing (4162017), and Youth Innovative Research Team of Capital Normal University.

### References

- [1] K. M. He, J. Sun, and X. O. Tang, "Single image haze removal using dark channel prior," in *Proceeding of Conference on Computer Vision and Pattern Recognition*, vol. 33, pp. 1956–1963, 2011.
- [2] L. Zeng and Y. Dai, "Single image dehazing based on combining dark channel prior and scene radiance constraint," *Journal of Electronics*, vol. 25, no. 6, pp. 1114–1120, 2016.
- [3] B. Li, S. Wang, J. Zheng, and L. Zheng, "Single image haze removal using content-adaptive dark channel and post enhancement," *IET Computer Vision*, vol. 8, no. 2, pp. 131–140, 2013.
- [4] X. Liu, H. Zhang, Y. Y. Tang, and J.-X. Du, "Scene-adaptive single image dehazing via opening dark channel model," *IET Image Processing*, vol. 10, no. 11, pp. 877–884, 2016.
- [5] B.-H. Chen, S.-C. Huang, and J. H. Ye, "Hazy image restoration by bi-histogram modification," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 4, article 50, pp. 1–17, 2015.
- [6] B.-H. Chen, S.-C. Huang, and F.-C. Cheng, "A high-efficiency and high-speed gain intervention refinement filter for haze removal," *Journal of Display Technology*, vol. 12, no. 7, Article ID 7384437, pp. 753–759, 2016.
- [7] S.-C. Huang, B.-H. Chen, and Y.-J. Cheng, "An efficient visibility enhancement algorithm for road scenes captured by intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2321–2332, 2014.
- [8] B.-H. Chen and S.-C. Huang, "Edge Collapse-Based Dehazing Algorithm for Visibility Restoration in Real Scenes," *Journal of Display Technology*, vol. 12, no. 9, Article ID 7450147, pp. 964–970, 2016.
- [9] S.-C. Huang, B.-H. Chen, and W.-J. Wang, "Visibility restoration of single hazy images captured in real-world weather conditions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 10, pp. 1814–1824, 2014.
- [10] Z. Luan, Y. Shang, X. Zhou, Z. Shao, G. Guo, and X. Liu, "Fast single image dehazing based on a regression model," *Neuro-computing*, vol. 245, pp. 10–22, 2017.
- [11] Z. Li and J. Zheng, "Edge-preserving decomposition-based single image haze removal," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5432–5441, 2015.
- [12] K. Tang, J. Yang, and J. Wang, "Investigating haze-relevant features in a learning framework for image dehazing," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 2995–3002, IEEE, Columbus, Ohio, USA, June 2014.
- [13] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao, "DehazeNet: an end-to-end system for single image haze removal," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5187–5198, 2016.
- [14] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang, "Single image dehazing via multi-scale convolutional neural networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9906, pp. 154–169, 2016.
- [15] M. Wang, J. Mai, Y. Liang, R. Cai, T. Zhengjia, and Z. Zhang, "Component-Based Distributed Framework for Coherent and Real-Time Video Dehazing," in *Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, pp. 321–324, Guangzhou, China, July 2017.
- [16] B. Li, X. Peng, and Z. Wang, "End-to-End United Video Dehazing and Detection, 2017."
- [17] J.-H. Kim, W.-D. Jang, J.-Y. Sim, and C.-S. Kim, "Optimized contrast enhancement for real-time image and video dehazing," *Journal of Visual Communication and Image Representation*, vol. 24, no. 3, pp. 410–425, 2013.
- [18] S. G. Narasimhan and S. K. Nayar, "Vision and the atmosphere," *International Journal of Computer Vision*, vol. 48, no. 3, pp. 233–254, 2002.
- [19] L. Anat, L. Dani, and W. Yair, "A closed-form solution to natural image matting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 228–242, 2015.
- [20] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.

## Research Article

# Efficient 3D Volume Reconstruction from a Point Cloud Using a Phase-Field Method

Darae Jeong <sup>1</sup>, Yibao Li,<sup>2</sup> Heon Ju Lee,<sup>3</sup> Sang Min Lee,<sup>3</sup> Junxiang Yang,<sup>1</sup> Seungwoo Park,<sup>1</sup> Hyundong Kim,<sup>1</sup> Yongho Choi <sup>1</sup>, and Junseok Kim <sup>1</sup>

<sup>1</sup>Department of Mathematics, Korea University, Seoul 02841, Republic of Korea

<sup>2</sup>School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China

<sup>3</sup>ROKIT Inc., Seoul 08512, Republic of Korea

Correspondence should be addressed to Junseok Kim; [cfdkim@korea.ac.kr](mailto:cfdkim@korea.ac.kr)

Received 22 November 2017; Accepted 1 January 2018; Published 6 February 2018

Academic Editor: Costică Moroşanu

Copyright © 2018 Darae Jeong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose an explicit hybrid numerical method for the efficient 3D volume reconstruction from unorganized point clouds using a phase-field method. The proposed three-dimensional volume reconstruction algorithm is based on the 3D binary image segmentation method. First, we define a narrow band domain embedding the unorganized point cloud and an edge indicating function. Second, we define a good initial phase-field function which speeds up the computation significantly. Third, we use a recently developed explicit hybrid numerical method for solving the three-dimensional image segmentation model to obtain efficient volume reconstruction from point cloud data. In order to demonstrate the practical applicability of the proposed method, we perform various numerical experiments.

## 1. Introduction

In this paper, we propose an efficient and robust algorithm for volume reconstruction from a point cloud. Reconstructing the three-dimensional model from a point cloud is important in medical applications. Surface reconstruction from a point cloud is a process of finding a surface model that approximates an unknown surface for a given set of sample points lying on or near the unknown surface [1].

Hoppe et al. developed an algorithm to reconstruct a surface in the three-dimensional space from unorganized points scattered on or near the unknown surface. The algorithm is based on the idea of determining the zero level set of a signed distance function [2]. Kazhdan proposed a surface reconstruction method which takes an oriented point set and returns a solid model. The method uses Stokes' theorem to calculate the characteristic function (one inside the model and zero outside of it) of the solid model [3]. To reconstruct implicit surfaces from scattered unorganized data set, Li et al. presented a novel numerical method for surface embedding narrow volume reconstruction from unorganized

points [4, 5]. Yang et al. proposed a 3D reconstruction technique from nonuniform point clouds via local hierarchical clustering [6]. Zhao et al. developed a fast sweeping level set and tagging methods [7]. Yezzi Jr. et al. proposed a new medical image segmentation based on feature-based metrics on a given image [8].

Beneš et al. used the Allen-Cahn equation with a forcing term to achieve image segmentation [9]. Caselles et al. proposed a model for active contours which could extract smooth shapes and could be adapted to find several contours simultaneously [10]. Methods using geometric active contour were introduced in [11–14]. Zhang et al. developed a weighted sparse penalty and a weighted grouping effect penalty in modeling the subspace structure [15]. Chen used an ICKFCM method (ICA analysis and KFCM algorithm) in medical image segmentation and made a good result in extracting the complicated images [16]. Zhang et al. proposed a novel fuzzy level set method based on finding the minimum of energy function to locate the true object boundaries effectively [17]. Other numerical studies based on level set method were also introduced in [18, 19].

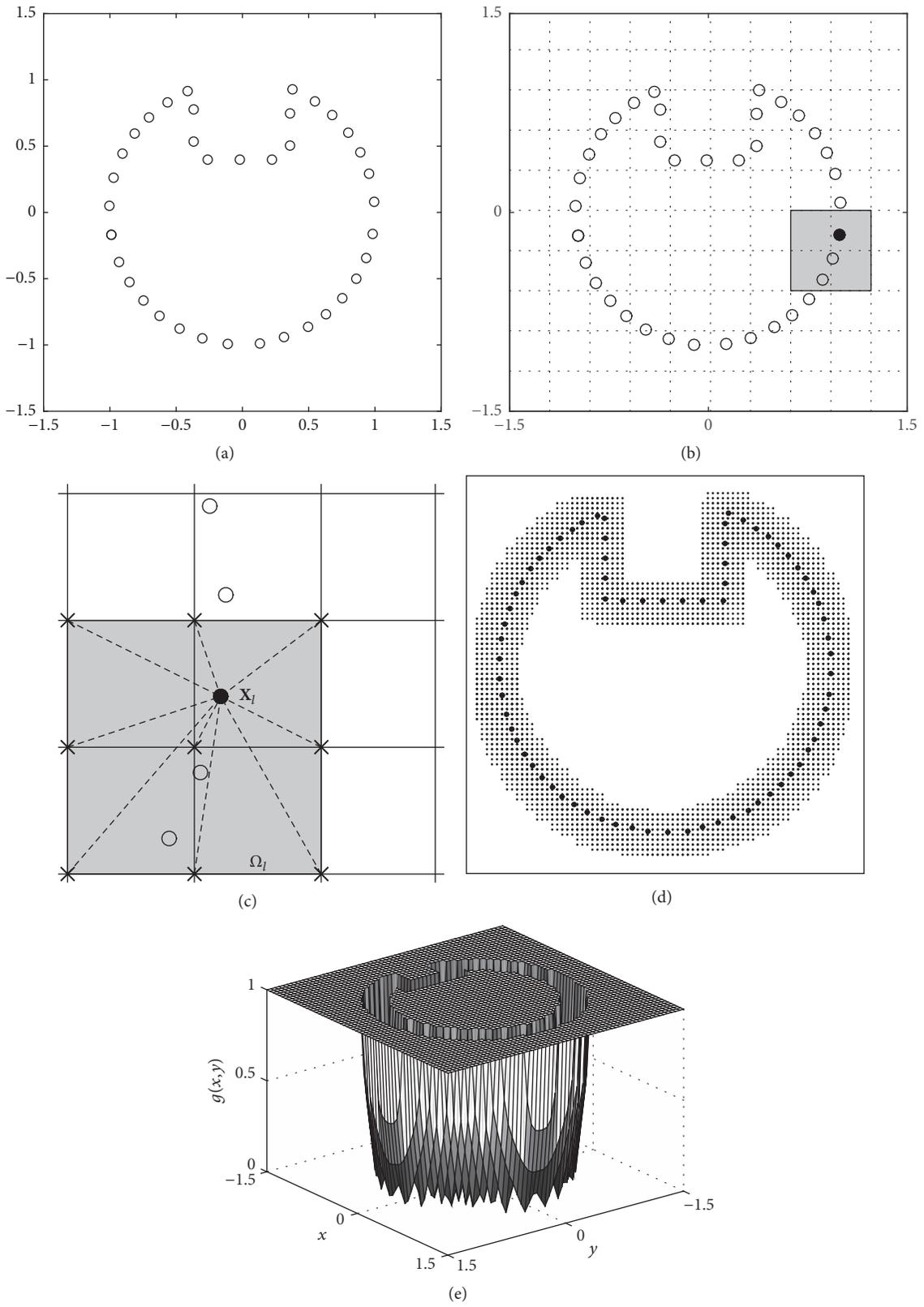


FIGURE 1: Schematic of defined distance function. (a) Point cloud data, (b) mesh grid covering the point cloud and local mesh grid, (c) local mesh grid  $\Omega_i$  embedding a point  $X_i$ , (d) narrow band domain, and (e) edge indicator function,  $g(x)$ .

In this article, we propose an explicit hybrid algorithm for volume reconstruction from a point cloud. Therefore, it does not need implicit solvers such as multigrid methods. The computation is fast and efficient because the proposed algorithm uses a narrow band domain and a good initial condition.

This paper is organized as follows. In Section 2, we describe a mathematical model and a numerical solution algorithm for volume reconstruction from a point cloud. We present the numerical results for several examples in Section 3. In Section 4, we conclude.

## 2. Mathematical Model and Numerical Solution Algorithm

Now, we propose an explicit hybrid numerical method for volume reconstruction from a point cloud using a phase-field method. For  $\mathbf{X}_l = (X_l, Y_l)$  in the two-dimensional space or  $\mathbf{X}_l = (X_l, Y_l, Z_l)$  in the three-dimensional space,  $S = \{\mathbf{X}_l \mid 1 \leq l \leq N\}$  denote the point cloud in the two- or the three-dimensional space, respectively. The geometric active contour model based on the mean curvature motion is given by the following evolution equation [20]:

$$\begin{aligned} \frac{\partial \phi(\mathbf{x}, t)}{\partial t} \\ = g(\mathbf{x}) \left[ -\frac{F'(\phi(\mathbf{x}, t))}{\epsilon^2} + \Delta \phi(\mathbf{x}, t) + \lambda F(\phi(\mathbf{x}, t)) \right], \end{aligned} \quad (1)$$

where  $g(\mathbf{x})$  is an edge indicator function,  $F(\phi) = 0.25(\phi^2 - 1)^2$ , and  $\epsilon$  is a constant which is related to the phase transition width. Note that here we use a different edge indicator function and efficient explicit numerical algorithm.

For simplicity of exposition, we first discretize (1) in the two-dimensional space  $\Omega = (a, b) \times (c, d)$ . Let  $h = (b - a)/(N_x - 1) = (d - c)/(N_y - 1)$  be the uniform mesh size, where  $N_x$  and  $N_y$  are the number of grid points. Let  $\Omega_h = \{\mathbf{x} = (x_i, y_j) : x_i = a + (i - 1)h, y_j = c + (j - 1)h, 1 \leq i \leq N_x, 1 \leq j \leq N_y\}$  be the discrete domain. Let  $\phi_{ij}^n$  be approximations of  $\phi(x_i, y_j, n\Delta t)$ , where  $\Delta t$  is the time step. Let  $d(\mathbf{x}) = \text{dist}(\mathbf{x}, S) = \min_{1 \leq l \leq N} |\mathbf{X}_l - \mathbf{x}|$  be the distance to the data  $S$ , where  $\mathbf{X}_l = (X_l, Y_l)$ . In fact, we will use the distance function as an edge indicator function,  $g(\mathbf{x})$ . In practice, for  $l = 1, \dots, N$ , we define a local domain  $\Omega_l$  which embeds the point  $\mathbf{X}_l$  and set the minimum value at the grid point between the point and the grid point. For example,  $\Omega_l$  is a  $3 \times 3$  grid. Then, the computational narrow band domain is defined as

$$\Omega_{\text{nb}} = \bigcup_{l=1}^N \Omega_l. \quad (2)$$

Outside the narrow band domain, we set a large value to the edge indicator function, see Figure 1 for the procedure.

In this study, we apply the simplest sequential splitting procedure. We split (1) into two equations by using the operator splitting method:

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = g(\mathbf{x}) [\Delta \phi(\mathbf{x}, t) + \lambda F(\phi(\mathbf{x}, t))], \quad (3)$$

$$\begin{aligned} \frac{\partial \phi(\mathbf{x}, t)}{\partial t} &= -g(\mathbf{x}) \frac{F'(\phi(\mathbf{x}, t))}{\epsilon^2} \\ &= g(\mathbf{x}) \frac{\phi(\mathbf{x}, t) - \phi^3(\mathbf{x}, t)}{\epsilon^2}. \end{aligned} \quad (4)$$

For a good initial configuration, we set  $\phi_{ij}^0 = 1$  inside on the narrow band domain and set  $\phi_{ij}^0 = -1$  outside the narrow band domain.

Given  $\phi^n$ , we solve (3) on the narrow band domain  $\Omega_{\text{nb}}$  by using the explicit Euler method:

$$\frac{\phi_{ij}^* - \phi_{ij}^n}{\Delta t} = g_{ij} [\Delta_d \phi_{ij}^n + \lambda F(\phi_{ij}^n)], \quad (5)$$

where  $\phi_{ij}^*$  is the intermediate value which is defined at point  $(x_i, y_j)$ . The initial values at inside and outside region of the narrow band domain  $\Omega_{\text{nb}}$  act as Dirichlet boundary condition in computing the discrete Laplace operator,  $\Delta_d$ .

Then, we analytically solve (4) by the method of separation of variables [21, 22]. That is,  $\phi_{ij}^{n+1} = \psi(\Delta t)$  by analytically solving

$$\frac{d\psi(t)}{dt} = g_{ij} \frac{\psi(t) - \psi^3(t)}{\epsilon^2} \quad (6)$$

with the initial condition  $\psi(0) = \phi_{ij}^*$ .

The analytic solution is given as

$$\phi_{ij}^{n+1} = \frac{\phi_{ij}^*}{\sqrt{e^{-2g_{ij}\Delta t/\epsilon^2} + (\phi_{ij}^*)^2 (1 - e^{-2g_{ij}\Delta t/\epsilon^2})}}. \quad (7)$$

Therefore, (5) and (7) consist of an efficient and robust algorithm for volume reconstruction from a point cloud. We should note that the proposed numerical solution algorithm is fully explicit. Therefore, we do not need an iterative method such as multigrid method to solve the governing equation. Also, the implementation of the algorithm is straightforward.

## 3. Computational Experiments

### 3.1. Two-Dimensional Experiments

**3.1.1. Motion by Mean Curvature.** To test the proposed numerical scheme, we perform a numerical experiment. The test is motion by mean curvature. If we set  $g(\mathbf{x}) = 1$  and  $\lambda = 0$ , then the governing equation (1) becomes the original Allen-Cahn equation [23], which is a reaction-diffusion equation describing the process of phase separation in a binary alloy

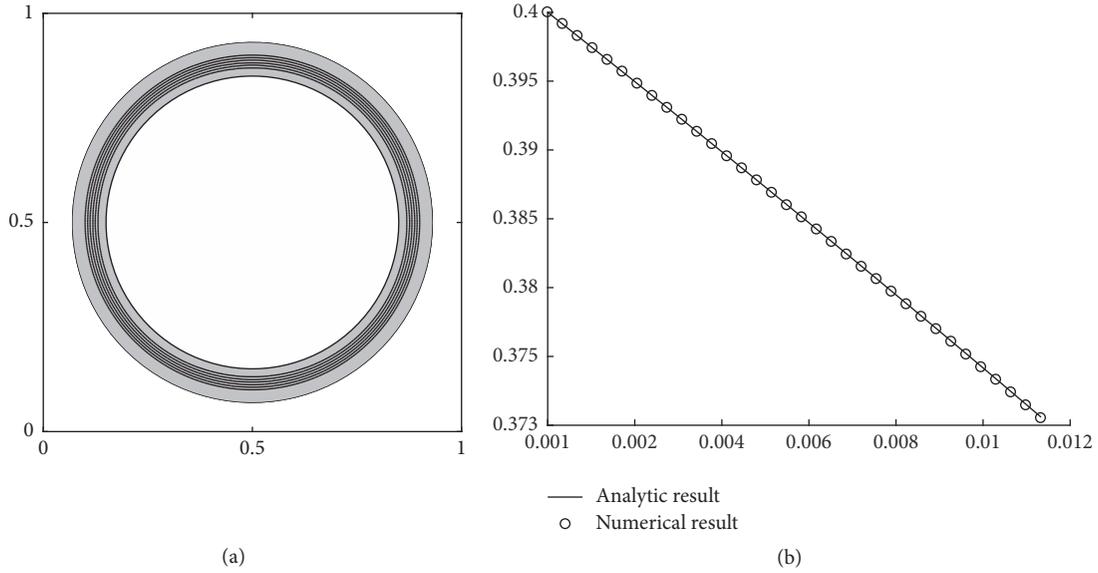


FIGURE 2: Temporal evolutions of the radius with  $\Delta t = 0.15h^2$  up to  $t = 5000\Delta t$  in the two-dimensional space. (a) Zero level contour and (b) radius  $R(t)$  of circle with respect to time.

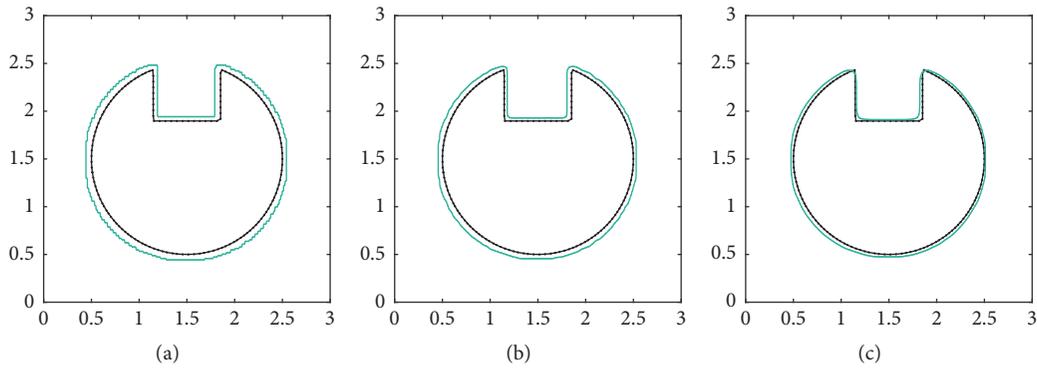


FIGURE 3: Temporal evolution of the interface: (a) initial condition, (b) 100 iterations, and (c) 1000 iterations.

mixture. In the two-dimensional case, as  $\epsilon$  approaches zero, the zero level set of  $\phi$  evolves with the following velocity:

$$V = -\kappa = -\frac{1}{R}, \quad (8)$$

where  $V$  is the normal velocity,  $\kappa$  is the curvature, and  $R$  is the radius of curvature at the point of the zero level set [24]. Then, (8) is rewritten by  $dR(t)/dt = -1/R(t)$  with  $R(0) = R_0$ . Therefore, analytic solution is given as  $R(t) = \sqrt{R_0^2 - 2t}$ .

On the computational domain  $\Omega = (0, 1) \times (0, 1)$ , we investigate the motion by mean curvature of the circle in the annulus narrow band domain:  $\{(x, y) \mid 0.35 \leq \sqrt{(x - 0.5)^2 + (y - 0.5)^2} \leq 0.43\}$  as shown in Figure 2(a). We define an initial condition as

$$\phi(x, y, 0) = \tanh \frac{R_0 - \sqrt{(x - 0.5)^2 + (y - 0.5)^2}}{\sqrt{2}\epsilon}. \quad (9)$$

In this numerical simulation, we use the following parameters:  $R_0 = 0.4$ ,  $\epsilon = 0.011$ ,  $h = 1/256$ ,  $\Delta t = 0.15h^2$ , and  $T = 5000\Delta t$ . Figures 2(a) and 2(b) show the temporal evolution of the initial circle and its radius with respect to time, respectively. For verification of our numerical results, we include the results of the analytic solution. As shown in Figure 2, the initial circle shrinks under the motion by mean curvature.

**3.1.2. The Basic Working Mechanism of the Algorithm.** The edge indicator function  $g(\mathbf{x})$  is close to zero where the point cloud exists. Therefore, the evolution will stop or slow down in the neighborhood of the point cloud. In (1),  $\partial\phi/\partial t = -F'(\phi)/\epsilon^2 + \Delta\phi$  makes the phase-field shrink until it reaches the point cloud by the mean curvature flow. If the geometry of the point cloud is not convex, then the term  $\lambda F(\phi)$  makes the level set of the phase-field further shrink. For more details, please refer to [20]. To confirm the working mechanism of the algorithm, the temporal evolution of the interface in the two-dimensional space is shown in Figure 3. Here, we use

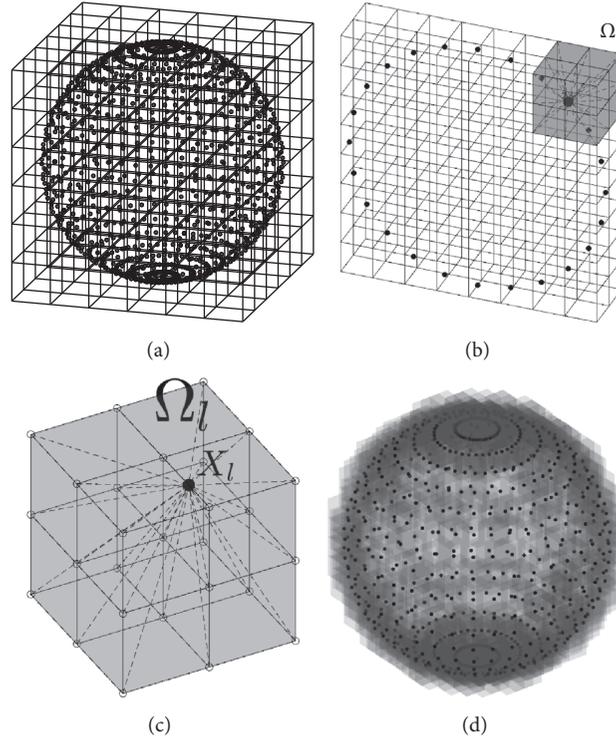


FIGURE 4: Construction of the three-dimensional distance function  $d(\mathbf{x})$  and the narrow band domain: (a) point cloud data and mesh, (b) cross section of point cloud and a local mesh  $\Omega_l$  of single point  $\mathbf{X}_l$ , (c) local mesh grid  $\Omega_l$  embedding a point  $\mathbf{X}_l$ , and (d) narrow band domain which is determined by the distance function.

the following parameters:  $\epsilon = 0.0096$ ,  $h = 1/200$ , and  $\Delta t = 2.8284e - 5$ .

**3.2. Three-Dimensional Experiments.** Next, we discretize (1) in the three-dimensional space, that is,  $\Omega = (a, b) \times (c, d) \times (e, f)$ . Let  $h = (b - a)/(N_x - 1) = (d - c)/(N_y - 1) = (f - e)/(N_z - 1)$  be the uniform mesh size, where  $N_x$ ,  $N_y$ , and  $N_z$  are the total number of grid points. Let  $\Omega_h = \{\mathbf{x} = (x_i, y_j, z_k) : x_i = a + (i - 1)h, y_j = c + (j - 1)h, z_k = e + (k - 1)h, 1 \leq i \leq N_x, 1 \leq j \leq N_y, 1 \leq k \leq N_z\}$  be the discrete domain. We define  $\phi_{ijk}^n$  as approximations of  $\phi(x_i, y_j, z_k, n\Delta t)$ , where  $\Delta t$  is the time step size. Let  $d(\mathbf{x}) = \text{dist}(\mathbf{x}, S) = \min_{1 \leq l \leq N} |\mathbf{X}_l - \mathbf{x}|$  be the distance to the data  $S$ , where  $\mathbf{X}_l = (X_l, Y_l, Z_l)$ .

Figure 4 represents construction of the three-dimensional distance function  $d(\mathbf{x})$  and the narrow band domain. In Figures 4(a) and 4(b), we can see the given point cloud data on computational grid and a local mesh  $\Omega_l$  of single point  $\mathbf{X}_l$ . Here, we calculate the distance between the given point  $\mathbf{X}_l$  and the grid points  $\mathbf{x}$  on the local mesh  $\Omega_l$ . The distance function  $d(\mathbf{x})$  is defined by the shortest one among the distance. Then, we obtain the narrow band domain which is determined by the distance function.

Now, we can straightforwardly extend the two-dimensional numerical solutions (5) and (7) to the following three-dimensional solutions:

$$\frac{\phi_{ijk}^* - \phi_{ijk}^n}{\Delta t} = g_{ijk} \left( \Delta_d \phi_{ijk}^n + \lambda F \left( \phi_{ijk}^n \right) \right),$$

$$\phi_{ijk}^{n+1} = \frac{\phi_{ijk}^*}{\sqrt{e^{-2g_{ijk}\Delta t/\epsilon^2} + \left(\phi_{ijk}^*\right)^2 \left(1 - e^{-2g_{ijk}\Delta t/\epsilon^2}\right)}}. \quad (10)$$

**3.2.1. Reconstruction from Various Point Clouds.** First, we reconstruct volume of Happy Buddha from the given scattered points ( $N = 1621848$ ) as shown in Figure 5(a) [25]. For numerical test, we use the following parameters:  $\epsilon = 0.0069$ ,  $h = 0.0048$ ,  $\Delta t = h/(200\sqrt{3})$ ,  $N_x = 250$ ,  $N_y = 500$ ,  $N_z = 250$ ,  $\Omega = (0, 1.2) \times (0, 2.4) \times (0, 1.2)$ , and  $\lambda = 100$ . In the first and second rows in Figure 5, we can see the front and back views of Happy Buddha. By the proposed scheme, we obtain the numerical solution  $\phi$  after 200 iterations (see Figure 5(c)) with the initial condition in Figure 5(b).

Next, we reconstruct volume of Armadillo model from the given scattered points ( $N = 129732$ ) as shown in Figure 6(a) [25]. For numerical test, we use the following parameters:  $\epsilon = 0.0127$ ,  $h = 0.0088$ ,  $\Delta t = h/(200\sqrt{3})$ ,  $N_x = 220$ ,  $N_y = 250$ ,  $N_z = 200$ ,  $\Omega = (0, 2.2) \times (0, 2.5) \times (0, 2)$ , and  $\lambda = 1000$ . In the first and second rows in Figure 6, we can see the front and back views of Armadillo model. By the proposed scheme, we obtain the numerical solution  $\phi$  after 400 iterations (see Figure 6(c)) with the initial condition in Figure 6(b).

As the final example, we reconstruct volume of Stanford Dragon from the given scattered points ( $N = 656469$ ) as shown in Figure 7(a) [25]. For numerical test, we use the



FIGURE 5: Front and back views of Happy Buddha: (a) point clouds, (b) initial condition of  $\phi$ , and (c) numerical solution  $\phi$  after 200 iterations.

following parameters:  $\epsilon = 0.00432$ ,  $h = 0.003$ ,  $\Delta t = h/(200\sqrt{3})$ ,  $N_x = 400$ ,  $N_y = 320$ ,  $N_z = 240$ ,  $\Omega = (0, 1.2) \times (0, 1) \times (0, 0.8)$ , and  $\lambda = 100$ . In the first and second rows in Figure 7, we can see the front and back views of Stanford Dragon. By the proposed scheme, we obtain the numerical solution  $\phi$  after 400 iterations (see Figure 7(c)) with the initial condition in Figure 7(b).

**3.2.2. Effect of  $\lambda$ .** In this section, we investigate the effect of  $\lambda$  parameter on the three-dimensional volume reconstruction. The parameter makes the level set of  $\phi$  shrink to the given points. We use the same parameters in Figure 7 except for the  $\lambda$  value. As shown in Figure 8, if the value of  $\lambda$  is small, then the surface is oversmoothed by the motion by mean curvature. On the other hand, if it is too large, then the surface is rough.

## 4. Conclusions

In this article, we developed an explicit hybrid numerical algorithm for the efficient 3D volume reconstruction from

unorganized point clouds using a modified Allen-Cahn equation. The 3D volume reconstruction algorithm is based on the 3D binary image segmentation method. The proposed algorithm has potential to be used in various practical industry such as 3D model printing from scattered scanned data. The computational results confirmed that the algorithm is very efficient and robust in reconstructing 3D volume from point clouds.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The first author (D. Jeong) was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIP) (NRF-2017R1E1A1A03070953). The corresponding author (Junseok Kim) was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry

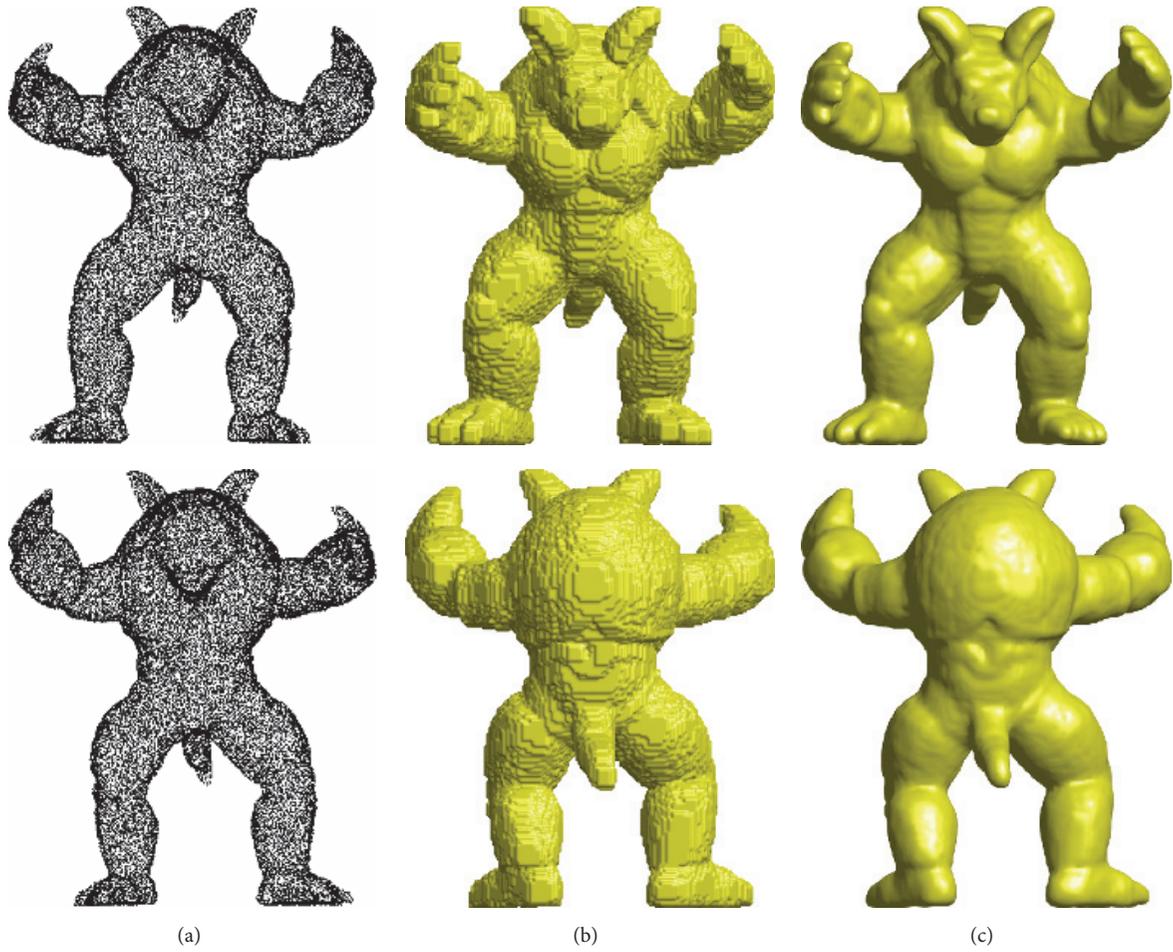


FIGURE 6: Front and back views of Armadillo model: (a) initial point clouds, (b) initial condition of  $\phi$ , and (c) numerical solution  $\phi$  after 400 iterations.

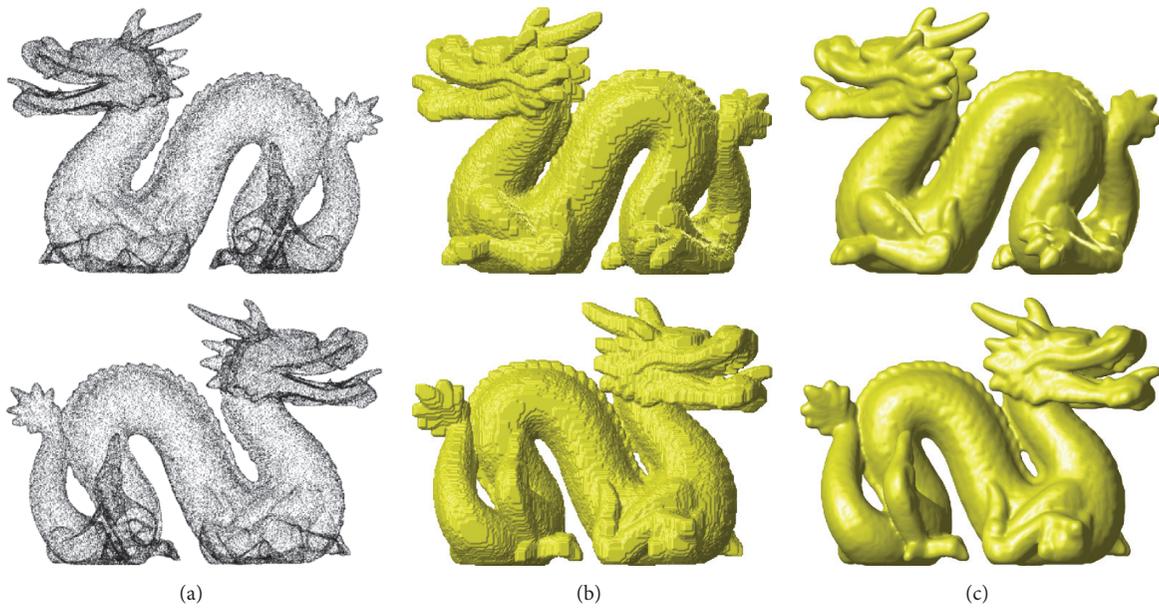


FIGURE 7: Front and back views of Stanford Dragon model: (a) initial point clouds, (b) initial condition of  $\phi$ , and (c) numerical solution  $\phi$  after 400 iterations.

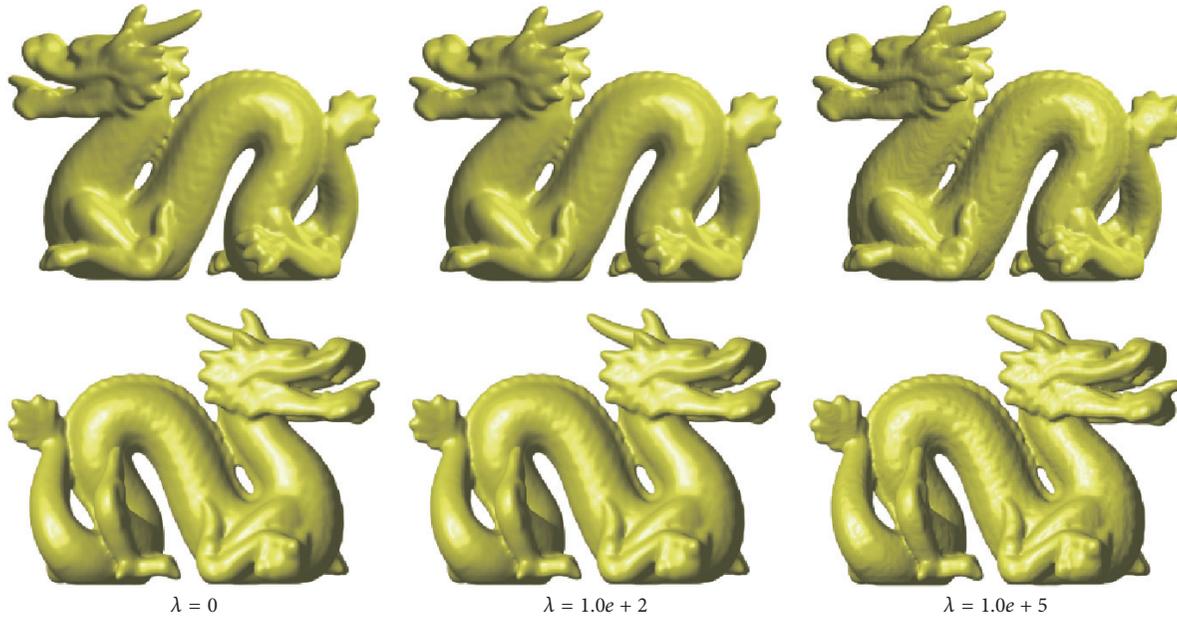


FIGURE 8: Front and back views of Stanford Dragon model with various  $\lambda$ . Numerical solution  $\phi$  after 1000 iterations.

of Education (NRF-2016R1D1A1B03933243). This work was supported by the BK21 PLUS program.

## References

- [1] F. Remondino, "From point cloud to surface: the modeling and visualization problem," in *Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXIV-5/W10, 2003.
- [2] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *Computer Graphics*, vol. 26, no. 2, pp. 71–78, 1992.
- [3] M. Kazhdan, "Reconstruction of solid models from oriented point sets," in *Proceedings of the third Eurographics Symposium on Geometry Processing*, pp. 73–82, 2005.
- [4] Y. Li, D. Lee, C. Lee et al., "Surface embedding narrow volume reconstruction from unorganized points," *Computer Vision and Image Understanding*, vol. 121, pp. 100–107, 2014.
- [5] Y. Li and J. Kim, "Fast and efficient narrow volume reconstruction from scattered data," *Pattern Recognition*, vol. 48, no. 12, article no. 5459, pp. 4057–4069, 2015.
- [6] J. Yang, R. Li, Y. Xiao, and Z. Cao, "3D reconstruction from non-uniform point clouds via local hierarchical clustering," in *Proceedings of the 9th International Conference on Digital Image Processing, ICDIP 2017, China, May 2017*.
- [7] H.-K. Zhao, S. Osher, and R. Fedkiw, "Fast surface reconstruction using the level set method," in *Proceedings of the IEEE Workshop on Variational and Level Set Methods in Computer Vision, VLSM 2001*, pp. 194–199, can.
- [8] A. Yezzi Jr., S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum, "A geometric snake model for segmentation of medical imagery," *IEEE Transactions on Medical Imaging*, vol. 16, no. 2, pp. 199–209, 1997.
- [9] M. Beneš, V. r. Chalupceky, and K. Mikula, "Geometrical image segmentation by the Allen-Cahn equation," *Applied Numerical Mathematics*, vol. 51, no. 2-3, pp. 187–205, 2004.
- [10] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," *Numerische Mathematik*, vol. 66, no. 1, pp. 1–31, 1993.
- [11] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.
- [12] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.
- [13] J. Hahn and C.-O. Lee, "Geometric attraction-driven flow for image segmentation and boundary detection," *Journal of Visual Communication and Image Representation*, vol. 21, no. 1, pp. 56–66, 2010.
- [14] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and J. Yezzi, "Conformal curvature flows: from phase transitions to active vision," *Archive for Rational Mechanics and Analysis*, vol. 134, no. 3, pp. 275–301, 1996.
- [15] B. Zhang, W. Wang, and X. Feng, "Subspace Clustering with Sparsity and Grouping Effect," *Mathematical Problems in Engineering*, vol. 2017, Article ID 4787039, 9 pages, 2017.
- [16] Y.-T. Chen, "Medical Image Segmentation Using Independent Component Analysis-Based Kernelized Fuzzy c-Means Clustering," *Mathematical Problems in Engineering*, vol. 2017, Article ID 5892039, 21 pages, 2017.
- [17] Y. Zhang, J. Xu, and H. D. Cheng, "A Novel Fuzzy Level Set Approach for Image Contour Detection," *Mathematical Problems in Engineering*, vol. 2016, Article ID 2602647, 12 pages, 2016.
- [18] C. Li, C. Xu, C. Gui, and M. D. Fox, "Level set evolution without re-initialization: a new variational formulation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 430–436, June 2005.
- [19] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the Mumford and Shah model," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 271–293, 2002.

- [20] Y. Li and J. Kim, "A fast and accurate numerical method for medical image segmentation," *Journal of the Korean Society for Industrial and Applied Mathematics*, vol. 14, no. 4, pp. 201–210, 2010.
- [21] A. M. Stuart and A. R. Humphries, *Dynamical Systems and Numerical Analysis*, vol. 2, Cambridge University Press, New York, NY, USA, 1998.
- [22] Y. Li, H. G. Lee, D. Jeong, and J. Kim, "An unconditionally stable hybrid numerical method for solving the Allen-Cahn equation," *Computers & Mathematics with Applications. An International Journal*, vol. 60, no. 6, pp. 1591–1606, 2010.
- [23] S. M. Allen and J. W. Cahn, "A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening," *Acta Metallurgica et Materialia*, vol. 27, no. 6, pp. 1085–1095, 1979.
- [24] B. Appleton and H. Talbot, "Globally optimal geodesic active contours," *Journal of Mathematical Imaging and Vision*, vol. 23, no. 1, pp. 67–86, 2005.
- [25] Stanford university computer graphics laboratory, <http://light-field.stanford.edu/acq.html>.

## Research Article

# Image Regularity and Fidelity Measure with a Two-Modality Potential Function

Weiwei Wang, Shengjiang Kong, Amir Razi, and Xiangchu Feng

School of Mathematics and Statistics, Xidian University, Xi'an 710126, China

Correspondence should be addressed to Weiwei Wang; [wwwang@mail.xidian.edu.cn](mailto:wwwang@mail.xidian.edu.cn)

Received 9 September 2017; Accepted 29 November 2017; Published 20 December 2017

Academic Editor: Gabriela Marinoschi

Copyright © 2017 Weiwei Wang 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 define a strictly convex smooth potential function and use it to measure the data fidelity as well as the regularity for image denoising and cartoon-texture decomposition. The new model has several advantages over the well-known ROF or TV- $L^2$  and the TV- $L^1$  model. First, due to the two-modality property of the new potential function, the new regularity has strong regularizing properties in all directions and thus encourages removing noise in smooth areas, while, near edges, it smoothes the edge mainly along the tangent direction and thus can well preserve the edges. Second, the new potential function is very close to the  $L^1$  norm; thus using it to measure the data fidelity makes the new model perform very well in removing impulse noise and preserving the contrast. Lastly, the proposed fidelity and regularization term is strictly convex and smooth and thus allows a unique global minimizer and it can be solved by using the steepest descent method. Numerical experiments show that the proposed model outperforms TV- $L^2$  and TV- $L^1$  in removing impulse noise and mixed noise. It also outperforms some state-of-the-art methods specially designed for impulse noise. Tests on cartoon-texture decomposition show that our method is effective and performs better than TV- $L^1$ .

## 1. Introduction

Image denoising aims to recover a clean image from a noisy observation. In this work, we mainly focus on removing the impulse noise, which randomly contaminates a portion of the pixels so that their true values are completely lost. The impulse noise is physically caused by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or transmission in a noisy channel [1]. It can be categorized into two types: one is the random-valued impulse noise, for which the noisy pixels can take any random values between the maximal and the minimal pixel values; and another is the salt-and-pepper noise, for which the noisy pixels can take only the maximal and minimal pixel values. For both types of noise, the noisy pixels are assumed to be randomly distributed in the image.

Let  $u(x, y) \in L^2(\Omega)$  be the original clean image defined on its domain  $\Omega \in R^2$ , with Lipschitz boundary and  $f(x, y) \in L^2(\Omega)$  be the observed image corrupted by impulse noise. The corruption can be formulated in the following general form:

$$f = N(u), \quad (1)$$

where  $N$  represents an impulse noise. Two main models for the impulse noise are used in a wide variety of applications: salt-and-pepper noise and random-valued impulse noise [2]. Denote the dynamic range of  $u$  by  $[d_{\min}, d_{\max}]$ ; that is,  $d_{\min} \leq u(x, y) \leq d_{\max}$ , for every pixel  $(x, y)$ , the model of the salt-and-pepper noise is defined by

$$f(x, y) = \begin{cases} d_{\min}, & \text{with probability } \frac{s}{2} \\ d_{\max}, & \text{with probability } \frac{s}{2} \\ u(x, y), & \text{with probability } 1 - s, \end{cases} \quad (2)$$

where  $f(x, y)$  denotes the gray level of  $f$  at a pixel location  $(x, y)$  and  $s$  determines the level of the salt-and-pepper noise. The model of the random-valued impulse noise is defined by

$$f(x, y) = \begin{cases} d(x, y), & \text{with probability } r \\ u(x, y), & \text{with probability } 1 - r, \end{cases} \quad (3)$$

where  $d(x, y)$  are identically and uniformly distributed random numbers in the range  $[d_{\min}, d_{\max}]$  and  $r$  defines the level of the random-valued impulse noise.

Image denoising is a typical ill-posed inverse problem and one of the most popular approaches is to solve a minimization problem of the form

$$\min_u D(u) + R(u), \quad (4)$$

where  $D(u)$  is a data fitting term derived according to the assumed noise type and  $R(u)$  is a regularization term that imposes the former on  $u$ . Many methods have been proposed by using various a priori knowledge about the image and the noise [3–5]. One of the most influential examples is the Rudin-Osher-Fatemi (ROF, or TV- $L^2$ ) [6]:

$$\min_u \int_{\Omega} (f - u)^2 dx dy + \lambda \int_{\Omega} |\nabla u| dx dy, \quad (5)$$

where  $|\nabla u|$  is the modulus of the gradient of  $u$ . The ROF model uses the  $L^2$  norm to measure the data fidelity under the additive Gaussian noise assumption and uses the total variation (TV) to measure the regularity by assuming the image is piecewise smooth or the gradient of the image is sparse. The total variation regularity allows for reconstruction of images with discontinuities across hypersurfaces and is extensively used in variational image restoration. Nevertheless, the  $L^2$  fidelity leads to some limitations. One important issue is the loss of contrast in the restored image even if the observed image is noise-free; another issue is that the fidelity term with  $L^2$  norm deals well with Gaussian noise but does not perform well in removing impulse noise. In [7], Chan and Esedoglu use the  $L^1$  norm as a measure of fidelity and formulate the following variational problem (TV- $L^1$ ):

$$\min_u \int_{\Omega} |f - u| dx dy + \lambda \int_{\Omega} |\nabla u| dx dy. \quad (6)$$

It was shown that the  $L^1$  norm better preserves the contrast, and the order in which features disappear in the regularization process is completely determined by their geometry (area and length), rather than the contrast as in the ROF model. This important geometric property is also used for the active contour global minimization problem [8]. Using the  $L^1$  fidelity, as analyzed in [7], model (6) implicitly detects the pixels contaminated by impulse noise and it preserves edges very well. Empirically, TV- $L^1$  outperforms TV- $L^2$  in detecting outliers and removing impulse noise [9]. However, in order to detect large noisy connected regions, it requires a greater weight of the regularization term in the cost function, which causes distortion of some pixels near edges. Moreover, it has some mathematical limitations: the minimizers of the variational problem (6) need not be unique in general because the  $L^1$  fidelity term is not strictly convex; it is not smooth either and solving the problem needs some regularization tricks. A weighted sum of  $L^1$  and  $L^2$  fidelities is used as the data fitting term and it works effectively and robustly for removal of mixed noise or almost any type of unknown noise

[10]. But it still suffers from the shortcomings of  $L^1$  fidelity. Huber norms [11] have been used for TV in order to avoid undesirable staircase effects [12]. In [13], the Huber loss

$$y = H_{\mu}(x) = \begin{cases} \frac{1}{2\mu}x^2, & |x| \leq \mu \\ |x| - \frac{\mu}{2}, & |x| > \mu \end{cases} \quad (7)$$

is used for both data fidelity and regularization. The advantage of using the Huber loss in comparison to the  $L^2$  norm is that geometric features such as edges are better preserved and it has continuous derivatives in contrast to the  $L^1$  norm that is not differentiable and leads to staircase artifacts. However, the Huber norm involves a parameter that affects the results.

Except the variational methods, some filtering based methods exist for impulse noise removal such as the Adaptive Median Filters (AMF) [14] and the Adaptive Center Weighted Median Filters (ACWMF) [15]. The AMF method uses Adaptive Median Filter with variable window size to filter out impulse noise. It is robust in removing mixed impulses with high probability of occurrence while preserving sharpness. But it is ineffective when an image is disturbed by other types of mixed noise, such as Gaussian, Poisson, and impulse noise. The ACWMF further uses spatial varying central weight to improve AMF and it is better than AMF in preserving details and in suppressing impulse noise, additive white noise, and signal dependent noise. However, the ACWMF tends to become an identity filter if impulses exist within a window and in that case, the ACWMF is not effective in suppressing impulses, especially for salt-and-pepper noise.

Cartoon-texture decomposition is an important mathematical tool for image analysis. It aims to decompose an image  $f$  into a cartoon component and a texture component. Ideally, the cartoon component is a piecewise smooth approximation of the original image and it mainly contains object hues and sharp edges while the texture component contains repeated small scale patterns. The general framework for cartoon-texture decomposition has the following from:

$$\begin{aligned} \min_{u,v} R(u) + \lambda D(v), \\ \text{s.t. } f = u + v, \end{aligned} \quad (8)$$

where  $R(u)$  and  $D(v)$  are two functionals, usually norms, measuring the cartoons  $u$  and  $v$ , respectively. Meyer [2] shows that the ROF model is ineffective in cartoon-texture decomposition because the  $L^2$  norm is not a good measure of the texture, yet the TV is effective in measuring the cartoon component. To overcome the ineffectiveness of the  $L^2$  norm in measuring the texture component, Meyer [2] and Haddad and Meyer [16] proposed using the  $G$ -norm, Vese and Osher [17] approximated the  $G$ -norm by the  $\text{div}(L^p)$  norm, Osher et al. [18] proposed using the  $H^{-1}$  norm, Lieu and Vese [19] proposed using the more general  $H^{-s}$  norm, and Le and Vese [20] proposed using the  $\text{div}(\text{BMO})$  norm to measure the texture component. However, the models involving these norms are difficult to solve. Yin et al. [21] show that the  $L^1$

norm is effective in measuring the texture component and proposed the TV- $L^1$  model.

In this work we define a strictly convex smooth potential function  $\phi(x)$  and use it to measure the data fidelity as well as the regularity for image restoration and cartoon-texture decomposition. Like Huber norm, the new potential function has two modalities: it is approximately half the square function (corresponding to the  $L^2$  norm) near 0 and approximately a linear function (corresponding to the  $L^1$  norm) when  $x$  is far away from 0. But Huber norm involves a parameter while our potential function does not. The new model has several advantages over the well-known Rudin-Osher-Fatemi (ROF) or TV- $L^2$  model and the TV- $L^1$  model. First, due to the two-modality property of the new potential function, using it working on the image gradient to measure the regularity makes the regularity work in two ways: in smooth area of the image, the regularity results in a diffusion term that is uniform and isotropic, having strong regularizing properties in all directions, and thus encourages removing noise in smooth area, while, near edges, the regularity results in a diffusion process which smooths the edge mainly along the tangent direction and thus can well preserve the edges. Such regularizing role of our regularity term is different from the TV; especially in smooth areas, TV regularity causes staircasing effect while our method does not. Second, the new potential function is very close to the  $L^1$  norm; thus using it to measure the data fidelity makes the new model perform very well in removing impulse noise and preserving the contrast. Lastly, the proposed fidelity and regularization term is strictly convex and smooth; thus the new model allows a unique global minimizer and it can be solved by using the steepest descent method. Mathematical analysis and numerical experiments show that the proposed model outperforms TV- $L^2$  and TV- $L^1$  in removing impulse noise and mixed noise. It also outperforms the Adaptive Median Filters (AMF) and the Adaptive Center Weighted Median Filters (ACWMF) in removing mixed noise. We also apply this model for cartoon-texture decomposition. Experimental results show it performs better than TV- $L^1$  in cartoon-texture decomposition.

## 2. The Proposed Model

The proposed model is as follows:

$$\min_u \int_{\Omega} \phi(f - u) dx dy + \lambda \int_{\Omega} \phi(|\nabla u|) dx dy, \quad (9)$$

where  $\lambda$  is a nonnegative tuning parameter and  $\phi(x)$  is defined by

$$\phi(x) = x \tan^{-1} x - \frac{1}{2} \ln(1 + x^2). \quad (10)$$

The rationality of this potential function can be explained as follows. First of all, the function  $\phi(x)$  is strictly convex and differentiable since  $\phi(x)' = \tan^{-1} x$  and  $\phi(x)'' = 1/(1 + x^2)$ ;

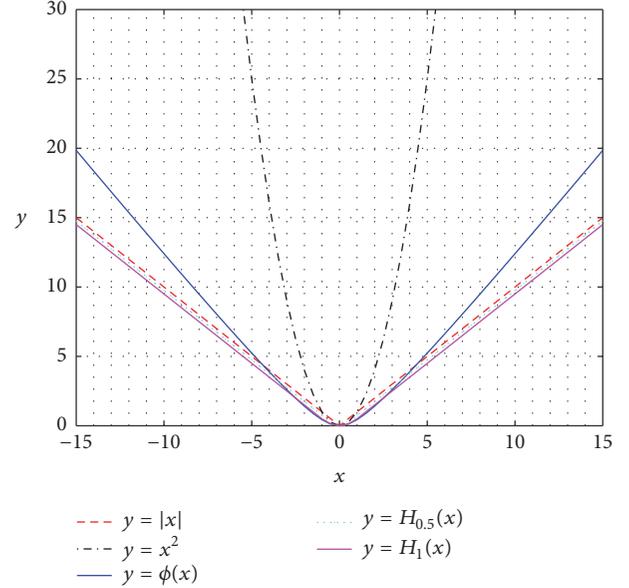


FIGURE 1: Graphs of four potential functions:  $y = |x|$ , corresponding to the  $L^1$  norm;  $y = x^2$ , corresponding to the  $L^2$  norm;  $y = \phi(x)$  corresponding to the proposed fidelity and regularization measure;  $H_{\mu}(x)$  corresponding to the Huber norm, where  $\mu = 0.5$  and  $\mu = 1$  are chosen.

thus our model (9) allows a unique global minimizer and it can be solved by using the steepest descent method. Secondly, when  $\phi(x)$  is used to measure the data fidelity, similar to Huber norm, it is also a good approximation of the  $L^1$  norm in the sense that  $\phi(x) \rightarrow x^2/2$  as  $x \rightarrow 0$  and  $\phi(x) \rightarrow (\pi/2)|x| - \ln|x|$  as  $|x| \rightarrow \infty$ , so it has similar performance to that of the  $L^1$  fidelity in removing impulse noise and preserving image contrast. Figure 1 compares  $\phi(x)$ ,  $L^1$  norm,  $L^2$  norm, and the Huber norm. Lastly, when  $\phi(x)$  is used to measure the regularity as in (9), it induces the following gradient descent flow:

$$\frac{\partial u}{\partial t} = \lambda \operatorname{div} \left( \frac{\phi'(|\nabla u|)}{|\nabla u|} \nabla u \right) + \phi'(f - u), \quad (11)$$

where the right side is the negative gradient of the functional in (9), with the first and the second term being deduced from the regularity term and the fidelity term, respectively. The diffusion term  $\operatorname{div}((\phi'(|\nabla u|)/|\nabla u|)\nabla u)$  can be decomposed as

$$\operatorname{div} \left( \frac{\phi'(|\nabla u|)}{|\nabla u|} \nabla u \right) = \frac{\phi'(|\nabla u|)}{|\nabla u|} u_{\mathbf{T}\mathbf{T}} + \phi''(|\nabla u|) u_{\mathbf{N}\mathbf{N}}, \quad (12)$$

where  $\mathbf{T}$  and  $\mathbf{N}$  denote the tangent and normal directions to the isophote lines (lines along which the intensity is constant) and  $u_{\mathbf{T}\mathbf{T}}$  and  $u_{\mathbf{N}\mathbf{N}}$  denote the second derivatives of  $u$  in the  $\mathbf{T}$ -direction and  $\mathbf{N}$ -direction. One can see that in a flat or smooth region of the image where the variations of the

intensity are weak, that is,  $t = |\nabla u| \approx 0$ , the coefficient of  $u_{\text{TT}}$ ,  $\phi'(t)/t$  and the coefficient of  $u_{\text{NN}}$ ,  $\phi''(t)$  satisfy

$$\begin{aligned} \lim_{t \rightarrow 0^+} \frac{\phi'(t)}{t} &= \lim_{t \rightarrow 0^+} \frac{\tan^{-1}(t)}{t} = \lim_{t \rightarrow 0^+} \frac{1}{1+t^2} = \lim_{t \rightarrow 0^+} \phi''(t) \\ &= 1 > 0; \end{aligned} \quad (13)$$

then (11) becomes

$$\frac{\partial u}{\partial t} = u_{\text{TT}} + u_{\text{NN}} + \phi'(f-u) = \Delta u + \phi'(f-u), \quad (14)$$

where  $\Delta$  denotes Laplacian differential operator. So, at these points,  $u$  locally satisfies (14), in which the diffusion term is uniform and isotropic, having strong regularizing properties in all directions, and thus encourages removing noise in smooth area. Near edges of the image, that is,  $t = |\nabla u| \gg 0$ ,  $\phi'(t)/t$ , and  $\phi''(t)$ , satisfy

$$\begin{aligned} \lim_{t \rightarrow +\infty} \frac{\phi'(t)}{t} &= \lim_{t \rightarrow +\infty} \frac{\tan^{-1}(t)}{t} = 0, \\ \lim_{t \rightarrow +\infty} \phi''(t) &= \lim_{t \rightarrow +\infty} \frac{1}{1+t^2} = 0, \\ \lim_{t \rightarrow +\infty} \frac{\phi''(t)}{\phi'(t)/t} &= \lim_{t \rightarrow +\infty} \frac{t}{(1+t^2)\tan^{-1}(t)} \\ &= \lim_{t \rightarrow +\infty} \frac{t}{1+t^2} \cdot \lim_{t \rightarrow +\infty} \frac{t}{\tan^{-1}(t)} = 0. \end{aligned} \quad (15)$$

This means the coefficient of  $u_{\text{TT}}$ ,  $\phi'(t)/t$  and the coefficient of  $u_{\text{NN}}$ ,  $\phi''(t)$  both vanish. However,  $\phi''(t)$  vanishes faster than  $\phi'(t)/t$ ; this allows the diffusion process to smooth the edge a little along the tangent direction; thus our regularity term can well preserve the edge. The TV regularity can be regarded as a special case of our regularity term by taking  $\phi(x) = x$ , then  $\phi'(x) = 1$ , and  $\phi''(x) = 0$ . In smooth area, that is,  $t = |\nabla u| \approx 0$ , the coefficient of  $u_{\text{TT}}$ ,  $\phi'(t)/t$  becomes large while the coefficient of  $u_{\text{NN}}$ ,  $\phi''(t) = 0$ ; this may be the reason why TV regularity causes the staircasing effect in smooth area.

The minimization problem (9) can be iteratively solved by the gradient descent method. Numerically, we use the following forward finite difference scheme to discrete the gradient descent flow (11):

$$u_{i,j}^{n+1} = u_{i,j}^n + \Delta t \cdot (\lambda B_{i,j}^n + \tan^{-1}(f_{i,j} - u_{i,j}^n)), \quad (16)$$

where  $\Delta t$  denotes the time step size and  $B$  is the diffusion term, defined by

$$\begin{aligned} B &= \text{div} \left( \frac{\phi'(|\nabla u|)}{|\nabla u|} \nabla u \right) \\ &= \frac{1}{1+|\nabla u|^2} \cdot \frac{u_x^2 u_{xx} + 2u_x u_y u_{xy} + u_y^2 u_{yy}}{|\nabla u|^2 + \varepsilon} \\ &\quad + \tan^{-1}(|\nabla u|) \cdot \frac{u_x^2 u_{yy} - 2u_x u_y u_{xy} + u_y^2 u_{xx}}{|\nabla u|^3 + \varepsilon} \end{aligned} \quad (17)$$

and  $\varepsilon$  is a regularizing constant to avoid dividing by 0, which is set by  $\varepsilon = 0.01$ , in our experiment. The spatial derivatives are discretized by central differences.

### 3. Numerical Simulation

This section is mainly devoted to numerical simulation of image denoising in the presence of impulse noise and mixed noise consisting of Gaussian, Poisson, and impulse noise. We also use our model to decompose an image into a cartoon component and a texture component. The simulations are performed using Matlab 8.5.0 (R2015a) in Windows 7 environment on 3.30 GHZ Intel Core i5-4590 CPU, 4 GB Ram PC. To assess the restoration performance quantitatively, we evaluate the peak signal to noise ratio (PSNR) defined as [22]

$$\text{PSNR} = 10 \lg \left( \frac{255^2}{(1/MN) \sum_{ij} (u_{i,j} - f_{i,j})^2} \right), \quad (18)$$

where  $u_{i,j}$  and  $f_{i,j}$  are the pixel values of the restored image and of the original image, respectively. In the presence of Poisson noise, the maximum intensity of the original noise-free image is varied in order to create images with different levels of Poisson noise.

**3.1. Image Denoising.** We first show the effectiveness of our method in removing impulse noise, including the salt-and-pepper noise and the random-valued impulse noise. In all experiments the time step size is set by  $\Delta t = 0.1$ .

The regularization parameter  $\lambda$  plays an important role in denoising because it balances the competition between the data fidelity and the regularization term. When  $\lambda$  takes large values, the regularization term dominates the total energy, which tends to force the restored image to be smoother and cleaner. When  $\lambda$  takes small values, the fidelity term dominates the total energy, which tends to force the restored image to be closer to the observed noisy image. In the following we analyze through experiments how the PSNR of the restored image depends on the value of  $\lambda$ . We show the results for the test images ‘‘Cameraman’’ (256 × 256) and ‘‘Lena’’ (256 × 256) with intensity values ranging from 0 to 255. In the experiment, the noisy images are produced by corrupting the test images with salt-and-pepper noise or random-valued impulse noise of levels 10%, 20%, and 30%. Figure 2 plots the PSNRs versus the values of  $\lambda$  for the image ‘‘Cameraman’’ with salt-and-pepper noise at different levels. Figure 3 plots that for random-valued impulse noise. From the plots, one can observe the following: first of all, in both cases of impulse noise, the PSNR of the three methods increases and reaches a maximum rapidly and then decreases slowly as the value of  $\lambda$  increases. Moreover, the optimal value (numerical) of  $\lambda$  (corresponding to the maximum PSNR) depends on the level of impulse noise. Lastly, for all levels of noise, the maximum PSNRs obtained by TV- $L^1$  and our method are comparative while the maximum PSNRs obtained by TV- $L^2$  are much lower (about 2 dB less). This again indicates that TV- $L^2$  is not fit for impulse noise removal.

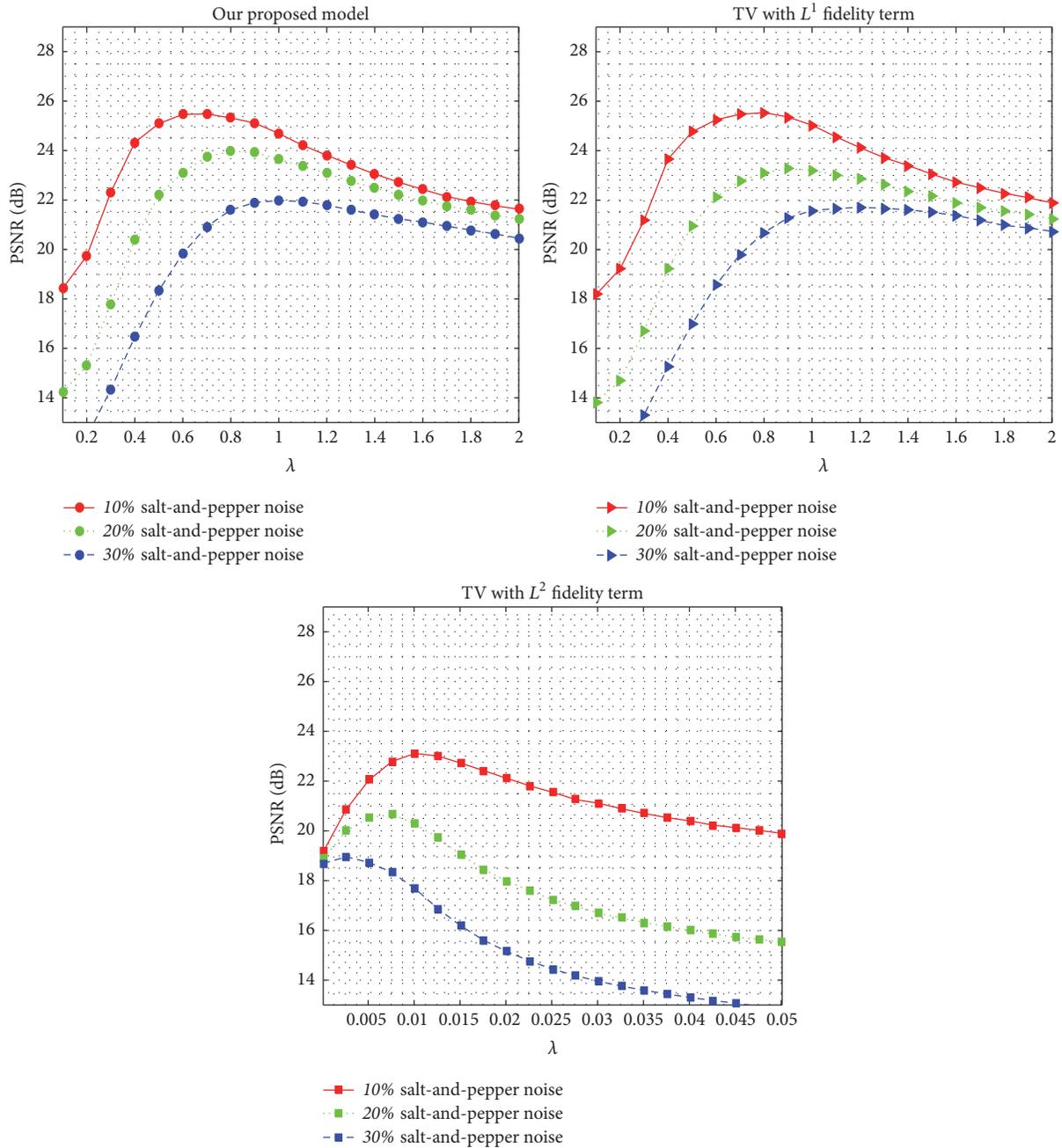


FIGURE 2: PSNR versus  $\lambda$  for the image “Cameraman” corrupted by salt and pepper noise at different levels.

To study how the optimal value of  $\lambda$  depends on the noise level for TV- $L^1$  and our method, we show some best values of  $\lambda$  corresponding to various levels of impulse noise in Figure 4, salt-and-pepper, and in Figure 5, random-valued. From Figures 4 and 5, one can see that TV- $L^1$  and our method have similar patterns of the dependency of the best  $\lambda$  on the noise level. In general, the higher the noise level, the larger the best value of  $\lambda$ . To be more specific, for both methods, the best value of  $\lambda$  tends to be stable in [0.8, 1.2] when the noise level is above 15%. Moreover, Figures 2 and 3 show that

the PSNR attenuates slowly if the value of  $\lambda$  is a little larger than the optimal value. For convenience, we choose  $\lambda = 1.1$  uniformly when the noise level is above 15% and  $\lambda = 0.8$  when the noise level is below 15%.

In the following experiments we compare visually and quantitatively the performance of our method with TV- $L^1$ , AMF, and ACWMF in removing impulse noise. Figures 6 and 7, respectively, show the results obtained by these methods for salt-and-pepper noise and random-valued impulse noise. The maximum window size used in AMF [14] is 19. The

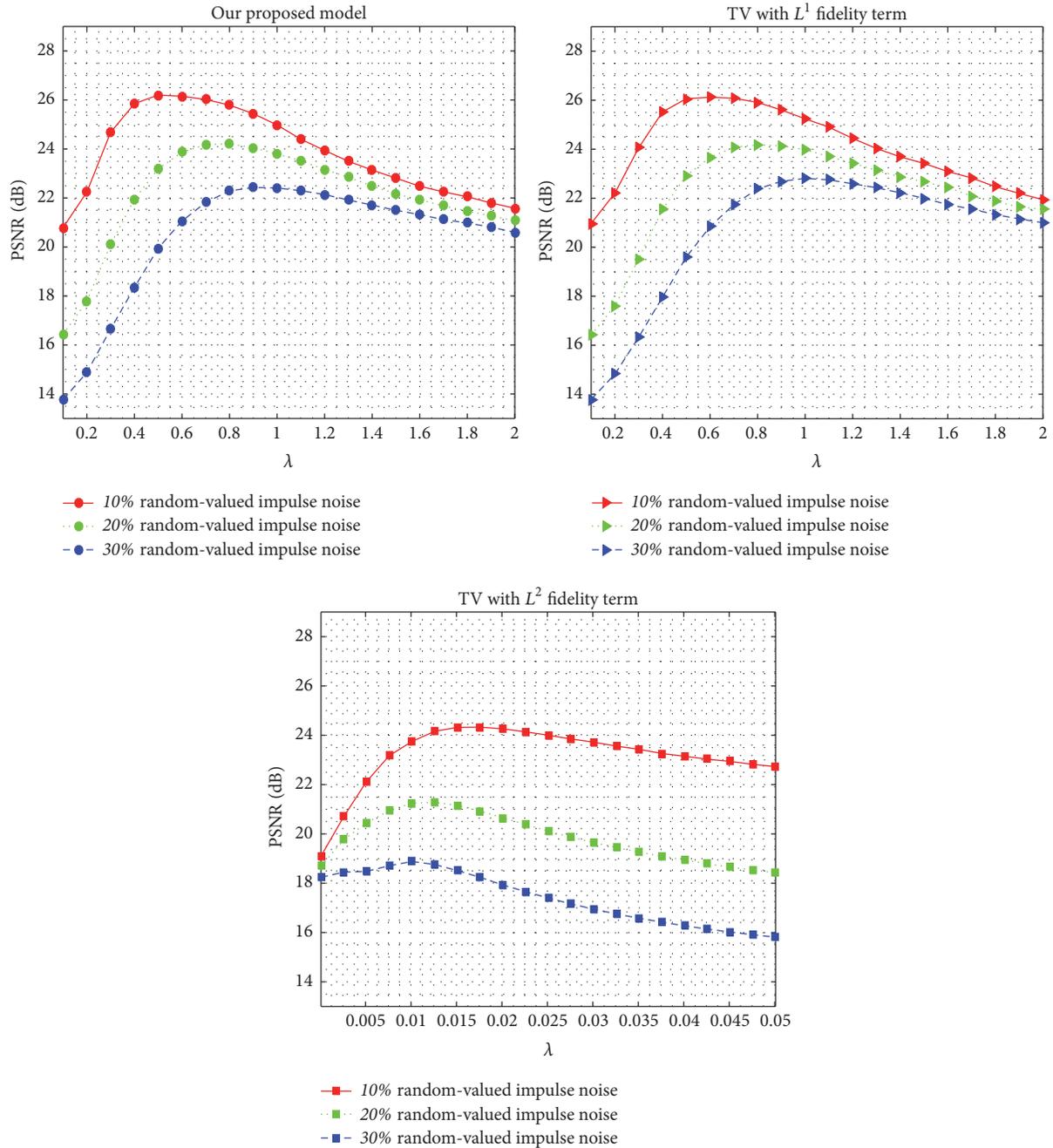


FIGURE 3: PSNR versus  $\lambda$  for the image “Cameraman” corrupted by random-valued impulse noise at different levels.

ACWMF [15] is successively performed 4 times with different parameters, which are chosen to be the same as those in [23]. Obviously, whether in removing salt-and-pepper noise or random-valued impulse noise, TV- $L^1$  and our method are well in removing noise and preserving the edges. But our method is a little better than TV- $L^1$  in two aspects. Objectively, the PSNR of our method is about 0.2~0.3 dB higher than that of TV- $L^1$ , and visually, there is less staircasing effect in the smooth area of the restored images. The ACWMF and AMF are better than TV- $L^1$  and our method in preserving

small scale details such as the textured ground in the image “cameraman,” and the PSNR of the AMF on the image “cameraman” is even higher than our method by 2.65 dB in case of salt-and-pepper noise. However, the ACWMF and AMF cannot successfully detect all the impulse noise in that some scattered peak points are visible in the restored images. Moreover, the AMF fails in suppressing the random-valued impulse noise.

As indicated in [10], the weighted sum of  $L^1$  and  $L^2$  fidelity is robust to any kind of commonly used noise prior, yet

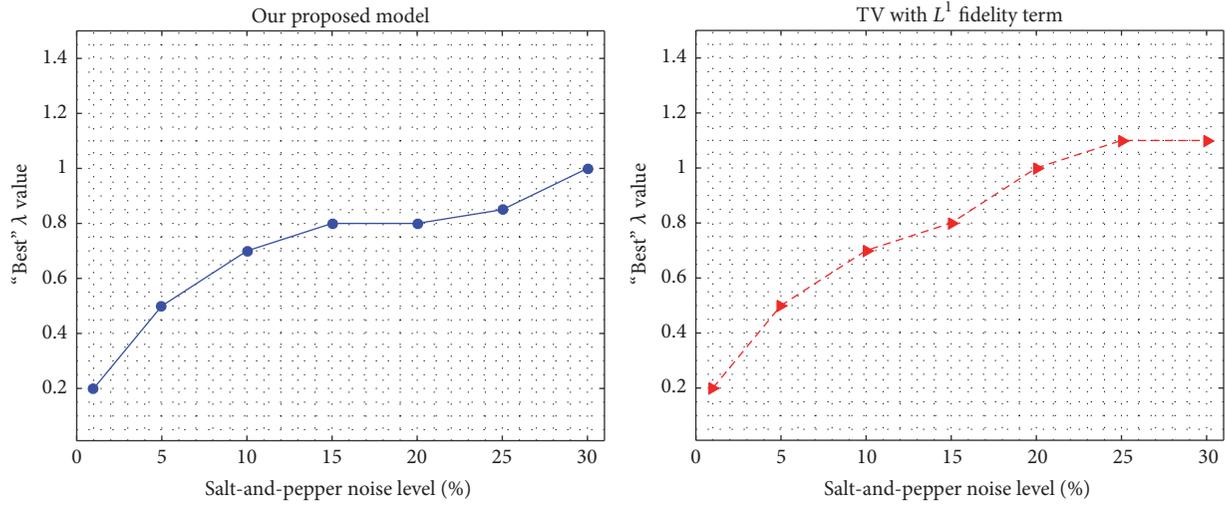


FIGURE 4: The best value of  $\lambda$  for the proposed method and TV- $L^1$  on the image “Cameraman,” in case of salt-and-pepper noise at different levels.

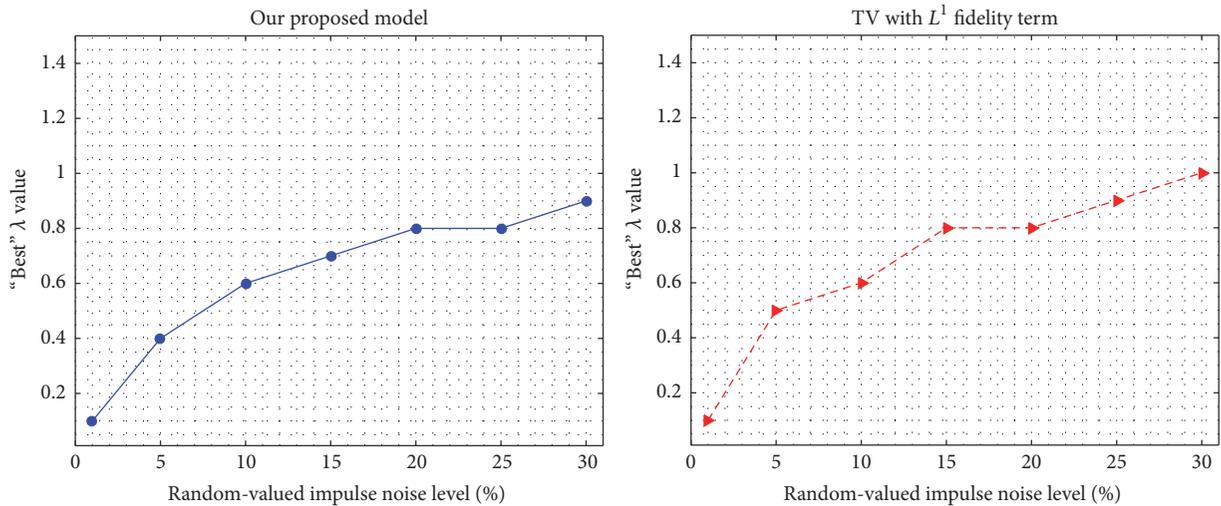


FIGURE 5: The best value of  $\lambda$  for the proposed method and TV- $L^1$  fidelity on the image “Cameraman,” in case of random-valued impulse noise at different levels.

empirically, the  $L^1$  norm absolutely dominates the fidelity. This motivates us to apply our model to remove Gaussian noise. Table 1 presents some results by TV- $L^2$ , TV- $L^1$ , and our method. Figure 8 compares the visual effects of these methods. One can see that TV- $L^1$  performs worse than the other two methods in case of higher level noise, and the restored image by TV- $L^2$  exhibits obvious and annoying staircase artifacts. The PSNRs and the restored images show that, for additive Gaussian noise, where the  $L^2$  fitting function is the best choice based on statistical analysis among all possible data fitting terms, our method performs better as well. The better results come from the two-modality property of our potential function.

Now we test the performance of our method in removing mixed noise consisting of additive Gaussian noise, Poisson noise, and impulse noise. We also compare our method with TV- $L^2$ , TV- $L^1$ , ACWME, and AMF. The Poisson noise is generated using the “*poissrnd*” function in Matlab with the input image scaled to the maximum intensity ( $I_{\max} = 255$ ). For the impulse noise, we only consider the random-valued impulse noise, because a pixel contaminated by such an impulse noise is not as distinctive as an outlier that is contaminated by salt-and-pepper noise and consequently is more difficult to detect. We consider three levels of the random-valued impulse noise: 10%, 20%, and 30%. The standard deviation of the white Gaussian noise is 10. For all

TABLE 1: Denoising results (PSNR) on three test images corrupted by Gaussian noise. The best PSNRs are given in bold.

Standard deviation $\sigma$	Boat ( $512 \times 512$ )			Lena ( $256 \times 256$ )			Rice ( $256 \times 256$ )		
	10	20	25	10	20	25	10	20	25
Our method	<b>31.77</b>	<b>28.42</b>	<b>27.98</b>	<b>30.75</b>	<b>27.98</b>	<b>27.15</b>	<b>31.97</b>	<b>29.15</b>	<b>28.38</b>
TV- $L^1$	31.40	28.32	27.50	30.47	27.60	26.50	31.61	27.80	27.64
TV- $L^2$	30.72	28.04	28.70	29.93	27.91	27.06	30.92	28.80	28.36



FIGURE 6: Image denoising results by different methods. (a) Original image; (b) corrupted “Cameraman” image with salt-and-pepper noise ( $s = 20\%$ , PSNR = 12.07 dB); (c) our method ( $\lambda = 1.1$ , PSNR = 25.82 dB); (d) TV- $L^1$  ( $\lambda = 1.1$ , PSNR = 25.58 dB); (e) ACWMF (PSNR = 25.23 dB); (f) AMF algorithm (PSNR = 28.47 dB).

cases, impulse noise is the first to be added and Gaussian noise is the last to be added. The PSNRs of different methods are presented in Table 2 and some of the restored images are shown in Figure 9. For all levels of impulse noise, our method obtains the best PSNRs and visual effects. TV- $L^1$  performs comparatively in removing impulse noise, but it does not perform as well as our method in removing mixture noise containing Gaussian noise. It may be explained by the two modalities of our potential function. The median filter based methods, especially the AMF is well fit for salt-and-pepper noise, but it does not perform well in case of random-valued impulse noise or mixed noise containing random-valued impulse noise. In fact, the AMF is good at detecting

salt-and-pepper noise because in that case, most of the noisy pixels are much more dissimilar to regular pixels and hence are easier to detect. However, the AMF is not effective in detecting random-valued impulse noise when the noise ratio is high.

**3.2. Cartoon-Texture Decomposition.** In this subsection, we show the effectiveness of our method in cartoon-texture decomposition and compare it with the TV- $L^1$  method. Since the function defined in (8) is a good approximation of the  $L^1$  norm, our model (9) can be used in cartoon-texture decomposition. We use model (9) to obtain  $u$  and finally take  $v = f - u$ .

TABLE 2: Denoising results (PSNR) on four test images in the presence of mixture of random-valued impulse noise, Gaussian noise with standard deviation  $\sigma = 10$ , and Poisson noise at image peak intensity of 255. The best PSNRs are given in bold.

Random-valued impulse noise ( $r$ ) + Poisson + Gaussian	Baboon ( $512 \times 512$ )			Plane ( $512 \times 512$ )			Bridge ( $512 \times 512$ )			Barbara ( $256 \times 256$ )		
	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%
Our method	<b>24.53</b>	<b>23.30</b>	<b>22.79</b>	<b>29.84</b>	<b>28.51</b>	<b>26.95</b>	<b>25.57</b>	<b>24.30</b>	<b>23.41</b>	<b>24.01</b>	<b>23.26</b>	<b>22.70</b>
TV- $L^1$	23.96	22.98	22.42	29.37	27.99	26.54	24.72	23.99	23.20	23.95	23.19	22.56
TV- $L^2$	21.24	20.97	20.65	24.76	23.52	21.91	21.51	20.82	19.89	21.35	20.71	19.93
ACWMF	23.65	23.01	22.43	26.24	25.47	24.50	24.55	23.80	22.87	22.87	22.35	21.77
AMF	23.21	20.84	18.93	23.56	20.29	17.93	23.12	20.19	17.88	22.52	20.03	18.03



FIGURE 7: Image denoising results by different methods. (a) Original image “Lena”; (b) corrupted image with random-valued impulse noise ( $r = 30\%$ , PSNR = 13.79 dB); (c) our method ( $\lambda = 1.1$ , PSNR = 27.02 dB); (d)  $TV-L^1$  ( $\lambda = 1.1$ , PSNR = 26.80 dB); (e) ACWMF (PSNR = 27.11 dB); (f) AMF (PSNR = 18.34 dB).

Figure 10 shows some results by the two methods on four test images, each of which contains smooth area bounded by large scale edges (cartoon) and repeated small scale details (texture). The top row shows the original test images. The other rows show the decomposition results. One can observe that our method can more thoroughly separate the cartoon and the texture. In the cartoon components obtained by  $TV-L^1$ , some textures are left behind. The cartoon component obtained by our method only contains the mainframe of the image, that is, the smoothed objects and their boundaries, and the small scale details are to a large extent separated into the texture part.

Finally we test the robustness of our method for cartoon-texture decomposition in presence of noise. The results are shown in Figure 11. The first row shows the input images: image (a) is corrupted with salt-and-pepper noise ( $s = 20\%$ ), Gaussian noise with standard deviation  $\sigma = 10$ , and Poisson noise; synthetic image (b) is corrupted with random-valued impulse noise ( $r = 20\%$ ), Gaussian noise with standard deviation  $\sigma = 10$ , and Poisson noise. Both  $TV-L^1$  and our method decompose the noise together with the texture. In the cartoon components obtained by  $TV-L^1$ , there exist noticeable staircase artifacts while the cartoon components obtained by our method are visually much better.

## 4. Conclusions

In this work we define a new potential function and use it to measure the data fidelity as well as the regularity for image denoising and cartoon-texture decomposition. The new potential function has some attractive mathematical properties: strictly convex, smooth, and two-modality, which makes the proposed model have some advantageous properties over the classical  $TV-L^2$  and  $TV-L^1$  models. For example, it can well remove wider categories of noise including additive Gaussian noise, impulse noise, Poisson noise, and their mixture; like TV regularity, it can well preserve important geometric structure such as image edges, but unlike TV regularity, it does not cause staircase effect in smooth areas; moreover, the new model allows a unique global minimizer and it can be solved by using the steepest descent method. Numerical experiments show that the proposed model outperforms  $TV-L^2$  and  $TV-L^1$  in removing commonly used noise. Tests on cartoon-texture decomposition show that our method is effective and performs better than  $TV-L^1$ .

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

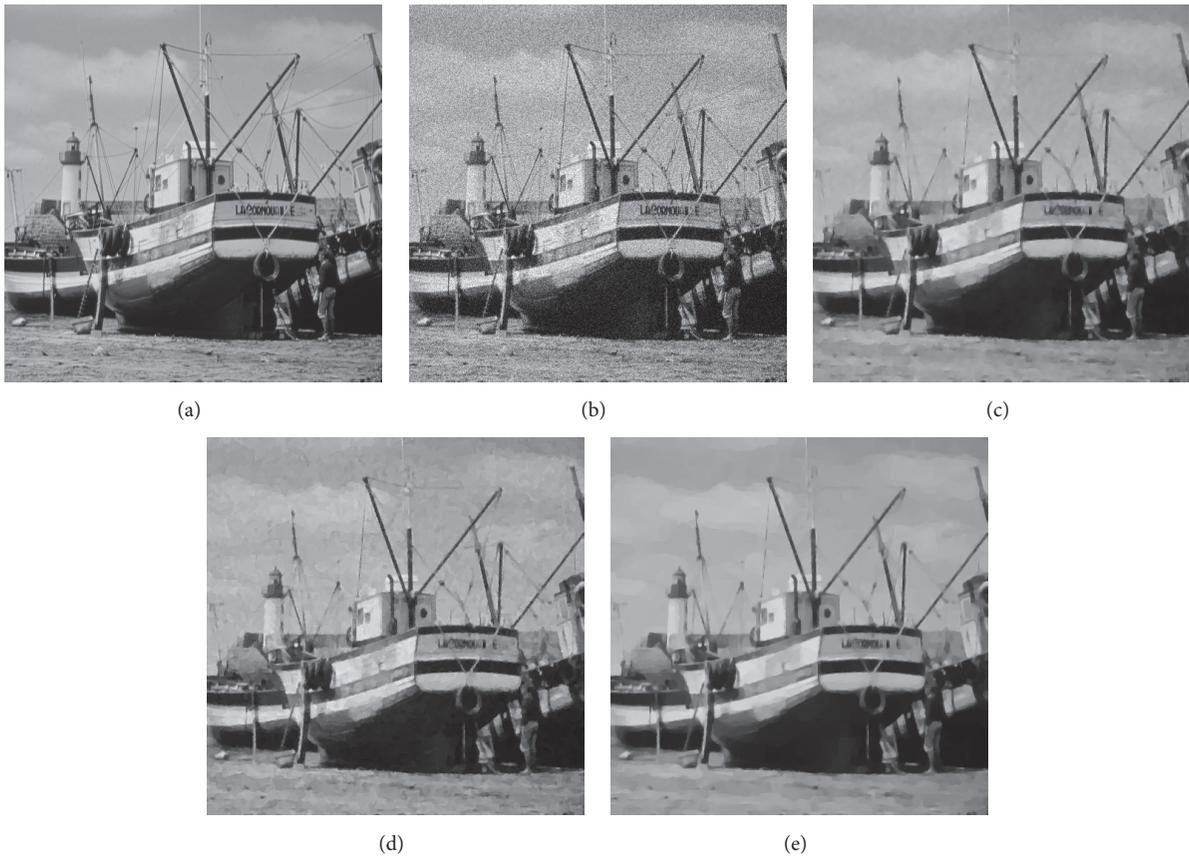


FIGURE 8: Denoising results of Gaussian noise by TV- $L^1$ , TV- $L^2$ , and our method. (a) Original test image “Boat”; (b) corrupted image with Gaussian noise (standard deviation  $\sigma = 20$ , PSNR = 22.14 dB); (c) our method ( $\lambda = 1.3$ , PSNR = 28.42 dB); (d) TV- $L^1$  ( $\lambda = 1.3$ , PSNR = 28.32), (e) TV- $L^2$  ( $\lambda = 0.01$ , PSNR = 28.04 dB).

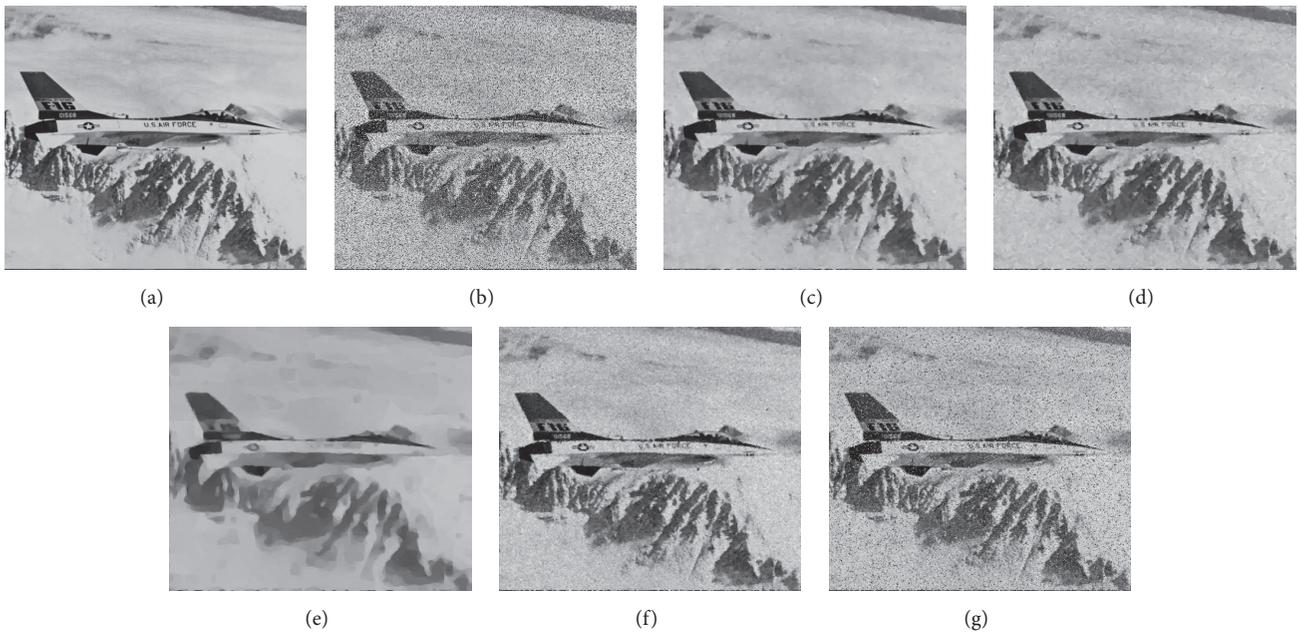


FIGURE 9: Denoising results of mixture noise of random-valued impulse noise, Poisson noise, and Gaussian noise by different methods. (a) Original test image “Plane”; (b) corrupted image with mixed noise ( $r = 30\%$ , PSNR = 13.56 dB); (c) our method ( $\lambda = 1.1$ , PSNR = 26.95 dB); (d) TV- $L^1$  ( $\lambda = 1.1$ , PSNR = 26.54 dB); (e) TV- $L^2$  ( $\lambda = 0.01$ , PSNR = 21.91 dB); (f) ACWM filter (PSNR = 24.50 dB); (g) AMF algorithm (PSNR = 17.93 dB).

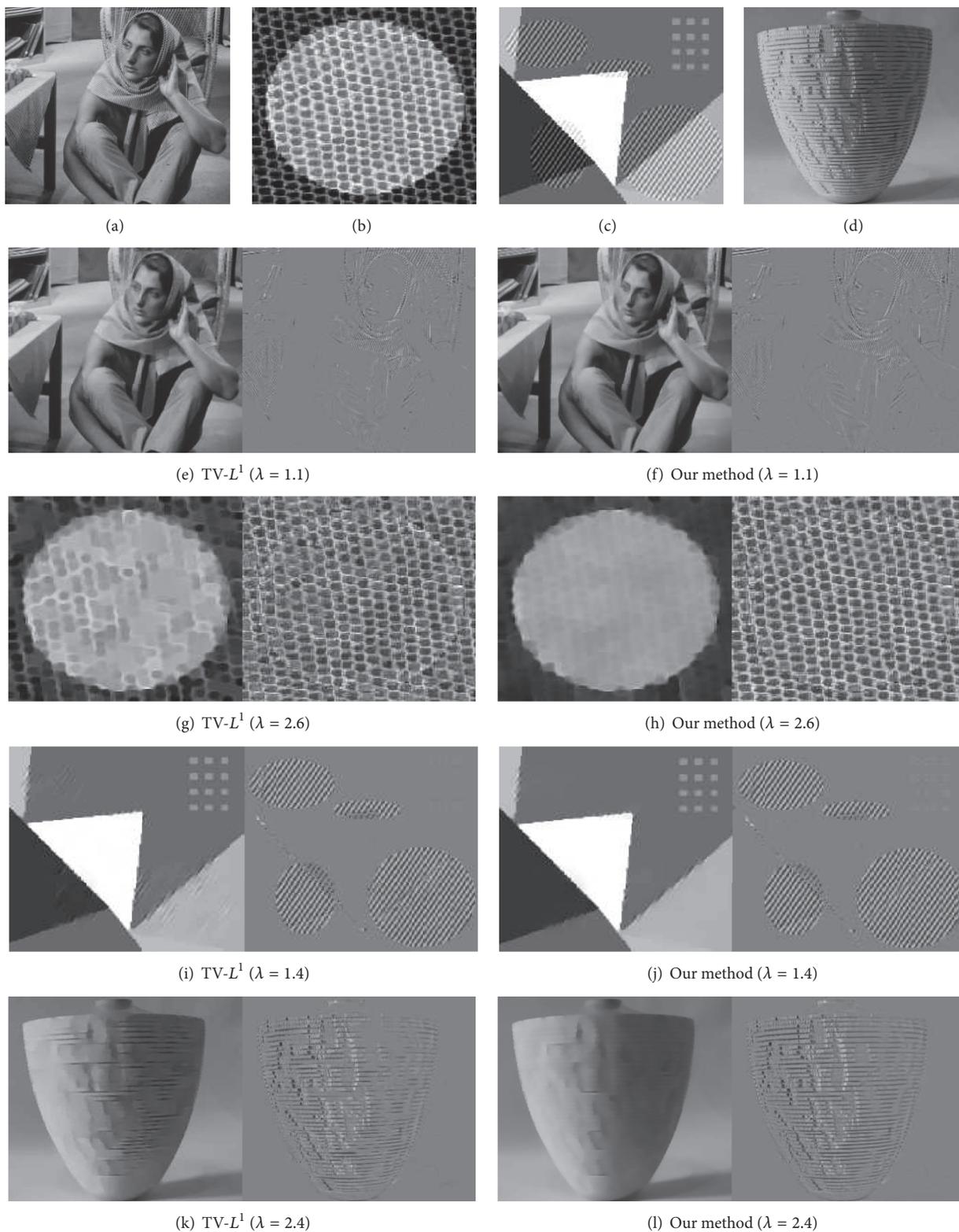


FIGURE 10: Cartoon-texture decomposition results (for each method, we show on the left the cartoon component and on right the texture).

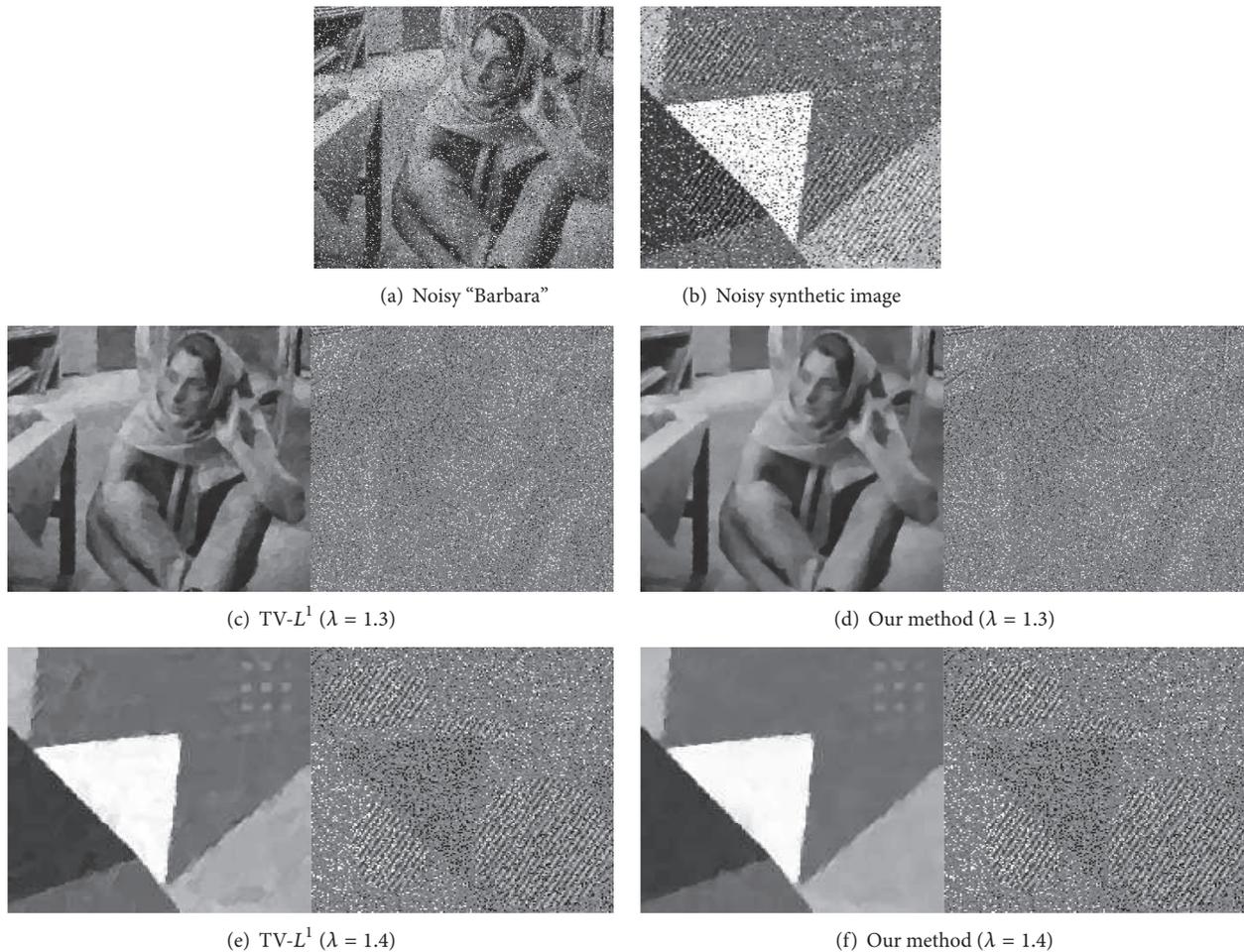


FIGURE 11: Cartoon-texture decomposition of images corrupted by mixed noise. Left: cartoon, right: texture.

## Acknowledgments

The authors would like to acknowledge the National Natural Science Foundation of China under Grants nos. 61472303, 61772389, and 61271294 and the Fundamental Research Funds for the Central Universities under Grant no. NSIY21 for supporting our research works.

## References

- [1] J. D. Gibson and A. Bovik, *Handbook of Image and Video Processing*, Academic Press, Cambridge, Mass, USA, 2000.
- [2] Y. Meyer, *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*, American Mathematical Society, Boston, Mass, USA, 2002.
- [3] S. Lefkimiatis, A. Roussos, M. Unser, and P. Maragos, "Convex generalizations of total variation based on the structure tensor with applications to inverse problems," in *Proceedings of the 4th International Conference on Scale Space and Variational Methods in Computer Vision (SSVM '13)*, vol. 7893 of *Lecture Notes in Computer Science*, pp. 48–60, Schloss Seggau, Austria, June 2013.
- [4] M. A. T. Figueiredo, J. B. Dias, J. P. Oliveira, and R. D. Nowak, "On total variation denoising: a new majorization-minimization algorithm and an experimental comparison with wavelet denoising," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '06)*, pp. 2633–2636, Atlanta, Ga, USA, October 2006.
- [5] C. R. Vogel and M. E. Oman, "Iterative methods for total variation denoising," *SIAM Journal on Scientific Computing*, vol. 17, no. 1, pp. 227–238, 1996.
- [6] 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.
- [7] T. F. Chan and S. Esedoglu, "Aspects of total variation regularized  $L^1$  function approximation," *SIAM Journal on Applied Mathematics*, vol. 65, no. 5, pp. 1817–1837, 2005.
- [8] T. F. Chan, S. Esedoglu, and M. Nikolova, "Algorithms for finding global minimizers of image segmentation and denoising models," *SIAM Journal on Applied Mathematics*, vol. 66, no. 5, pp. 1632–1648, 2006.
- [9] M. Nikolova, "A variational approach to remove outliers and impulse noise," *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1-2, pp. 99–120, 2004.
- [10] Z. Gong, Z. Shen, and K.-C. Toh, "Image restoration with mixed or unknown noises," *Multiscale Modeling and Simulation*, vol. 12, no. 2, pp. 458–487, 2014.
- [11] P. J. Huber, "Robust estimation of a location parameter," *Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.

- [12] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.
- [13] B. Hong, J. Koo, M. Burger, and S. Soatto, "Adaptive regularization of some inverse problems in image analysis," <https://arxiv.org/abs/1705.03350>.
- [14] H. Hwang and R. A. Haddad, "Adaptive median filters: new algorithms and results," *IEEE Transactions on Image Processing*, vol. 4, no. 4, pp. 499–502, 1995.
- [15] S.-J. Ko and Y. H. Lee, "Center weighted median filters and their applications to image enhancement," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 38, no. 9, pp. 984–993, 1991.
- [16] A. Haddad and Y. Meyer, "Variational methods in image processing," *Contemporary Mathematics*, vol. 446, 2007.
- [17] L. A. Vese and S. J. Osher, "Modeling textures with total variation minimization and oscillating patterns in image processing," *Journal of Scientific Computing*, vol. 19, no. 1-3, pp. 553–572, 2003.
- [18] S. Osher, A. Solé, and L. Vese, "Image decomposition and restoration using total variation minimization and the  $H^1$ ," *Multiscale Modeling & Simulation*, vol. 1, no. 3, pp. 349–370, 2003.
- [19] L. H. Lieu and L. A. Vese, "Image restoration and decomposition via bounded total variation and negative Hilbert-Sobolev spaces," *Applied Mathematics & Optimization*, vol. 58, no. 2, pp. 167–193, 2008.
- [20] T. M. Le and L. A. Vese, "Image decomposition using total variation and  $\text{div}(\text{BMO})$ ," *Journal of Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 390–423, 2005.
- [21] W. Yin, D. Goldfarb, and S. Osher, "Image cartoon-texture decomposition and feature selection using the total variation regularized  $\ell_1$  functional," in *Variational, Geometric, and Level Set Methods in Computer Vision*, vol. 3752 of *Lecture Notes in Computer Science*, pp. 73–84, Springer, Berlin, Germany, 2005.
- [22] 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.
- [23] R. H. Chan, C. Hu, and M. Nikolova, "An iterative procedure for removing random-valued impulse noise," *IEEE Signal Processing Letters*, vol. 11, no. 12, pp. 921–924, 2004.