Robust Interactive Segmentation Using Color Histogram and Contourlet Transform

Guoheng Huang and Chi-Man Pun

Abstract—**Efficient and accurate image segmentation is an important task in computer vision and object recognition. Since fully automatic image segmentation is hard to handle with natural images and texture images with complex background, thus interactive scheme with a few simple user inputs is a very good addition to image segmentation. For the purpose to accurately extract objects from different images, this paper presents a color histogram and Contourlet transform based interactive image segmentation. In the initialization stage, a superpixel based initial segmentation is applied to the original image. After that, the original image will be divided into a certain number of superpixels. Then, each superpixel will be represented by a novel superpixel feature based on color histogram and Contourlet transform. Finally, by using user-defined strokes, we merge the superpixels which are similar to marked object superpixels and merge the superpixels similar to marked background superpixels, until classify all superpixels. The computational complexity is analyzed, and comparative experimental results show that the proposed scheme can reliably and rapidly extract the desired object from the complex background.**

Index Terms—**Interactive segmentation, color histogram, superpixel, Contourlet transform, region merging.**

I. INTRODUCTION

The main purpose of image segmentation is to exact the edge of the interested objects from background. There are generally two mainstream image segmentations: one is automatic segmentation; on the contrary, the user input or initial parameter is necessary for the second kind of image segmentation - interactive segmentation. In general, fully automatic segmentation is easy to operate and high in speed. However, background regions of some images are so complex that it is very difficult for fully automatic segmentation to extract object contour from background. Therefore, interactive segmentation is proposed for such segmentation tasks. Recently, more and more research papers are focusing on interactive segmentation. Therein, a Graph Cuts based technique for general purpose interactive segmentation was put forward by Boykov and Jolly that can be also applied to mark certain pixels as "object" or "background" to provide hard constraints for segmentation [1]-[3]. For instance, Grab Cut and Grow Cut are two popular implements of Graph Cuts like algorithms [4]-[6]. Random Walks method is also another popular algorithm of interactive segmentation. An algorithm based on Random Walks for image segmentation was first proposed by Grady

[7] in 2006, and then a large amount of Random Walks deformation one after another [8]-[10]. Moreover, Level Set is also another popular algorithms applied to interactive segmentation [11]-[13]. In addition, there are still many kinds of algorithm which are specific [14]-[22]. Recently, an interactive segmentation based on region merging (MSRM) is proposed by Ning [23]. The thought of this algorithm was to divide the original image into some regions by Mean-shift, and then merge the object regions and the background regions individually. However, MSRM is not robust enough to segment texture images because it does not consider the texture information of image. Nevertheless, it also inspires us to simplify the segmentation process by initial segmentation.

In this paper, we apply superpixel like clustering SLIC as the initial segmentation at the beginning. After the first stage, original image is divided into a desired number of size-equal superpixels. Then, the remaining task is to classify by object and background label. Moreover, for the purpose to represent each superpixel, the feature extraction of superpixel is necessary. Hence we use the combination of color histogram and Contourlet transform as superpixel feature for merging. Influent experiments verified that this new algorithm was better than some references and it is also effective to segment texture images.

The remaining parts of this paper are arranged as follow. In Section II, we will introduce the related background and theory. In Section III, we will give an overview of our algorithm first, and then introduce every step. Final, we will present experimental results and compare our algorithm with other algorithms, and then conclude this paper in Section IV and V.

II. RELATED WORK

A. Simple Linear Iterative Clustering

SLIC (Simple Linear Iterative Clustering) was proposed by S üsstrunk [24]. It is a fast and efficient superpixel like method. It is an improved method of k-means for superpixel generation, and it has two improvements: first, the number of optimized distance calculations is observably reduced by limiting the search space to a region proportional to the size of superpixel that reduces the complexity to be linear in the number of pixels N and independent of the number of superpixels k; second, a weighted distance measure which includes color and spatial proximity while providing control over the size and density of the superpixels at the same time.

The only default parameter of the algorithm is k, which is the desired number of superpixels. For the CIELAB color model, the SLIC procedure begins from centers of the initial

cluster $C_i = [l_i\ a_i\ x_i\ y_j]\ T$, $i = 1\ldots k$. For the purpose to produce roughly equally sized superpixels and the pixel distance is $\sqrt{N/K}$. To reduce the chance of seeding a superpixel with a noisy pixel, the centers are moved to seeds according to the lowest gradient position in a 3×3 neighborhood (Fig. 1).

The quantities of superpixels $k = 100, 200$ and 300.

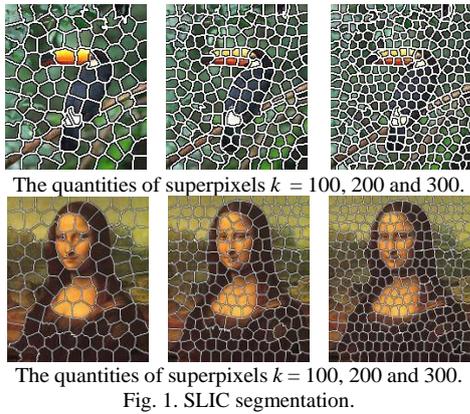The quantities of superpixels $k = 100, 200$ and 300.
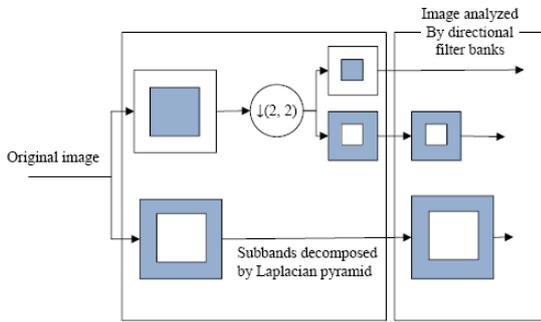Fig. 1. SLIC segmentation.

Fig. 2. The procedure of contourlet transform.

Then, each pixel i is associated with the nearest cluster center whose search region overlaps its location. This is the key point to speeding up SLIC because of limiting the size of the search region significantly reduces the number of distance calculations, and results in a significant speed advantage over conventional k-means clustering where each pixel must be compared with all cluster centers. This is only possible through the introduction of a distance measure D, which determines the nearest cluster center for each pixel. Since the expected spatial extent of a superpixel is a region of size $\sqrt{N/K} \times \sqrt{N/K}$, the search for similar pixels is done in a region $2\sqrt{N/K} \times 2\sqrt{N/K}$ around the superpixel center.

When each pixel has been associated with the nearest cluster center, an update would adjust the cluster centers to be the mean vector of all the pixels which belong to the cluster.

B. Contourlet Transform

A filter bank structure is what can deal with piecewise smooth images with smooth contours, was proposed by Do and Vetterli [25]. Contourlet can overcome the challenges of Wavelet and Curvelet transform. Contourlet transform is double filter bank structure. It is implemented by the Pyramidal Directional Filter Bank (PDFB) which decomposes images into directional subbands at multiple scales. In terms of structure, the Contourlet transform is a combination of Laplacian Pyramid and Directional Filter Bank. On the whole, it first uses a Wavelet-like transform for edge detection, and then a local directional transform for

contour segment detection.

Conceptually, the flow chart of Contourlet transform can be illustrated by Fig. 2, where the Laplacian Pyramid iteratively decomposes a 2-D image into low pass and high pass subbands, and the DFB is applied to the high pass subbands to further decompose the frequency spectrum. Moreover, the actual Contourlet transform is shown in Fig. 3.
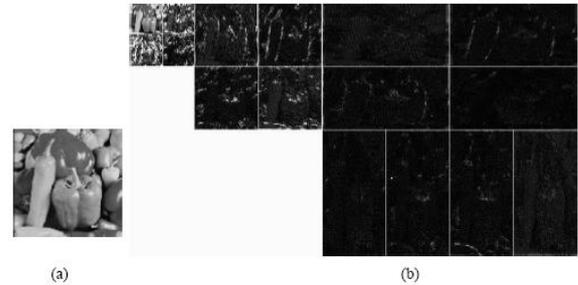

Fig. 3. Two-level Contourlet transform. (a) Original image. (b) One level of Wavelet transform and two levels of Pyramidal decompositions.
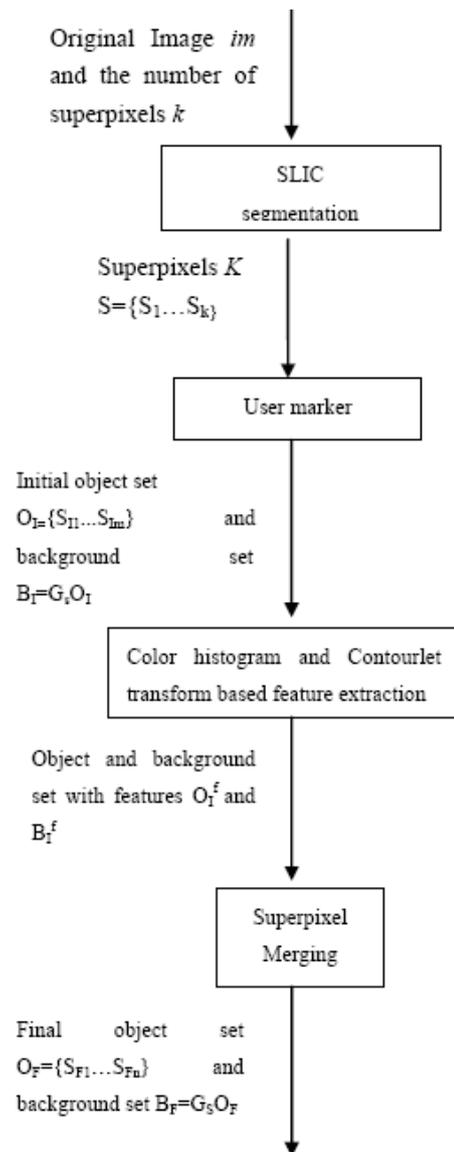
III. PROPOSED METHOD

Fig. 4. The procedure of proposed algorithm.

In our method as shown Fig. 4, an initial segmentation is required to divide the image into a desired number of superpixels for merging. In fact, the initial segmentation has an effect on the final segmentation result. There are many existing algorithms can cater this case, such as superpixel, Mean-shift and Watershed which are potential for this step. Original image is easy to be over-segmented and divided into too much regions by classical superpixel and Watershed. In this paper, we will apply a Clustering-like theory based on superpixel SLIC for initial segmentation. As mentioned, the image will be divided into many superpixels $S = \{s_1 \dots s_k\}$. Next, user need to mark object and background superpixels and obtain initial object superpixel set $O_I = \{s_{I1} \dots s_{Im}\}$ and background superpixel set $B_I = \complement_S O_I$. At this moment, for the purpose to divide the image into two regions - object and background, the remaining work is to merge object and background superpixels individually. Before that, the feature definition of each superpixel is necessary. In [23], Color histogram was chosen as the superpixel feature. However, Color histogram only considers the color information each region but the frequency information. It cannot obtain positive segmentation result when segment texture image. Therefore, a novel superpixel feature based on color histogram and Contourlet transform that considers both color information and frequency information for each superpixel is proposed in this paper.

A. Initial Segmentation

As mentioned before, SLIC is a k-means clustering algorithm based on superpixel. Thus, its procedure is similar to normal k-means clustering. After input the original image, we must segment the image initially, and divide it into some superpixels. In this method, the first thing must be done is to provide the initial parameter - the amount of the superpixels. The bigger the number is, the more superpixels will be divided; the lesser the number is, the less superpixels will be divided. If too many superpixels are divided, it takes much time on merging; if too few superpixels are divided, it is difficult to extract the exact edge for each superpixel. Therefore, we must choose proper number of superpixels. Of course, in practice use in the future, we will continue to research an adaptive method to find the number of super pixels according to the size of image or the frequency distribution of image.

In this paper, we will divide the original image *im* into a desired number of superpixels $S = \{s_1 \dots s_k\}$ at the beginning.

B. Feature Extraction

After initial segmentation, image was divided into some equal-size superpixels. In order to guide the superpixel merging process, these superpixels must be represented by using some descriptor and define a rule for merging. In [23], the region was described in the color histogram which is an effective descriptor to represent the object color feature. However, color histogram has its limitation because it only considers the color information of each superpixel but the frequency information. It is not effective when segments texture image. Therefore, we put forward a new superpixel similarity feature constructed by color histogram and Contourlet transform $\rho(s_i, s_j)$, where s_i and s_j are two different

superpixels.

For superpixel s_i, denoted by

$$fCoHist(S_i) = \frac{\sum_{u=1}^{4096} Hist_{s_i}^u}{norm(\sum_{u=1}^{4096} Hist_{s_i}^u)}$$

Especially, $f_{CoHist}(s_i)$ is a unit vector. Where $Hist_{s_i}$ is the normalized histograms of s_i, and the upperscript u represents its uth element.

Second, we propose the other superpixel feature based on Contourlet transform. At this moment, each superpixel is irregular region. Thus, we extend the irregular region to its binary enclosing square (such as 128×128). Then, Contourlet transform would be applied to each binary enclosing square of each superpixxel s_i $i = 1 \dots k$, where k is the number of superpixels. After Contourlet transform (in this paper, it includes one level Wavelet and two level Pyramidal decomposition), then decompose 10 components $C_{ij}, j = 1 \dots 10$. Where i is the sequence of each superpixel, $C_{ij}, j = 1 \dots 4$, is the component of Wavelet transform, the remaining components are from two level Pyramidal decomposition. Calculate their normalized energy and let them compose 10-dimensional vector (norm (C_{i1})…norm (C_{i10}). Unitize these vectors to be 10-dimensional unit vectors $f_{Contourlet}(S_i) = \frac{(norm(C_{i1)},\dots, norm(Ci10))}{norm((norm(C_{i1)},\dots, norm(C_{i10)))}}$.

The superpixel similarity measure for color histogram p_{CoHist} between S_i and S_j will be defined as $p_{CoHist}(S_i, S_j) = f_{CoHist}(s_i) \cdot f_{CoHist}(s_j)$, which is within [0, 1]. The superpixel similarity measure for Conourlet $p_{Contourlet}$ between S_i and S_j is defined as $p_{Contourlet}(s_i, s_j) = f_{Contourlet}(s_i) \cdot f_{Contourlet}(S_j)$, which is also within [0, 1]. Therefore, p is denoted by $p = w_1 \cdot p_{CoHist}(s_i, s_j) + w_2 \cdot p_{Contourlet}(s_j)$, where w_1 and w_2 are weights. The closer that ρ is to 1, the more similar these two superpixels may turn out to be.

C. Superpixel Merging

For the merging procedure, we can refer to [23]. The whole process is a two-stage process, which are repeatedly executed until no new merging occurs: once merging all the background superpixels together, the remaining is the object superpixels.

In summary, the first step of our method is to divide the original image into superpixels by SLIC. Then, we compute the color histogram and Contourlet transform based superpixel feature for each superpixel. Based on these superpixel features, we merge the object superpixels and background superpixels individually.

IV. ANALYSIS AND RESULTS

In this section, we have implemented our proposed interactive method on the environment: Intel Core i7-3517U 1.90GHz CPU with 4GB RAM and applied it to fluent testing images including natural images and texture images. Our method is essentially a semi-automatic segmentation method. With the strokes marked by the user, it automatically merge superpixels and label the non-marker superpixels as object or background. Therefore, we need to provide the initial user marker and ratio parameter between color histogram feature and Contourlet feature. Next, we will individually discuss the performance of our method in natural images and texture images and compare with other traditional algorithms: RWR (Random Walks with Restart) [8], User-Friendly algorithm (Segmentation through Unified User Input) [26] and SLRW (Segmentation with Multiple Linear Reconstructions in Windows) [27].

A. Tests for Natural Images

In this section, the experimental objects are natural images with fluent colors. Therefore, in this part of tests, we use 0.9 as the weight of color histogram feature and 0.1 as the weight of Contourlet feature. First, we use SLIC algorithm as the initial segmentation; second, we mark the object in green and background in blue. Visual comparison results are shown by Fig. 5 to Fig. 9.
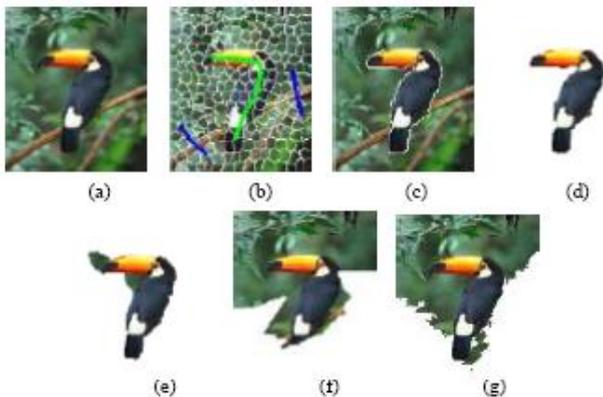

Fig. 5. The first comparison for natural image. (a) Bird. (b) User marker and SLIC segmentation. (c) Object boundary. (d) Proposed method. (e) RWR. (f) User-Friendly. (g) SLRW.
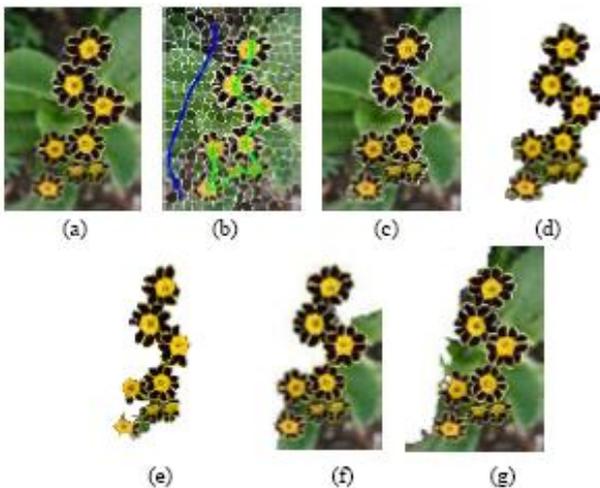

Fig. 6. The second comparison for natural image. (a) Flower. (b) User marker and SLIC segmentation. (c) Object boundary. (d) Proposed method. (e) RWR. (f) User-Friendly. (g) SLRW.
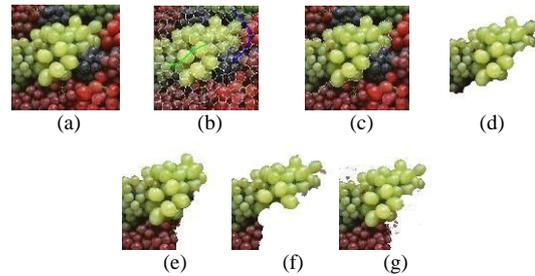

Fig. 7. The third comparison for natural image. (a) Grape. (b) User marker and SLIC segmentation. (c) Object boundary. (d) Proposed method. (e) RWR. (f) User-Friendly. (g) SLRW.
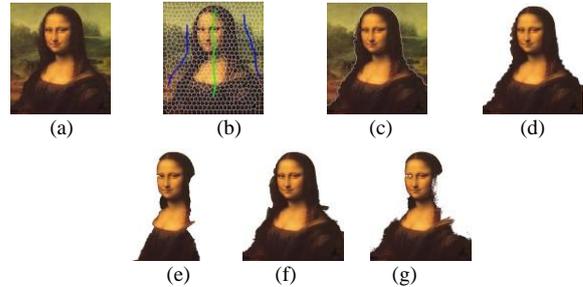

Fig. 8. The fourth comparison for natural image. (a) Mona Lisa. (b) User marker and SLIC segmentation. (c) Object boundary. (d) Proposed method. (e) RWR. (f) User-Friendly. (g) SLRW.
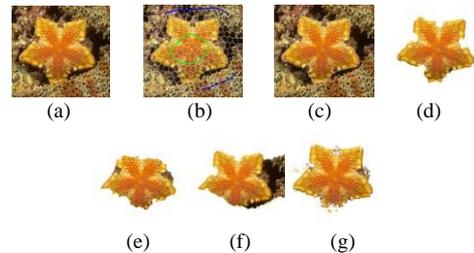

Fig. 9. The fifth comparison for natural image. (a) Starfish. (b) User marker and SLIC segmentation. (c) Object boundary. (d) Proposed method. (e) RWR. (f) User-Friendly. (g) SLRW.

From above figures, we can observe that our method can achieve the better results. These experimental results confirm the effectiveness to extract the objects of natural images. Therein, it cannot exact a positive boundary for object by RWR and User-Friendly; there are some discrete points in the SLRW result. In general, our algorithm gives the relatively positive results

B. Tests for Texture Images

In this section, the experimental objects are gray-level images with different textures. Therefore, we use 0.1 as the weight of color histogram feature and 0.9 as the weight of Contourlet feature. Visual comparison results are shown by Fig. 10 to Fig. 12.
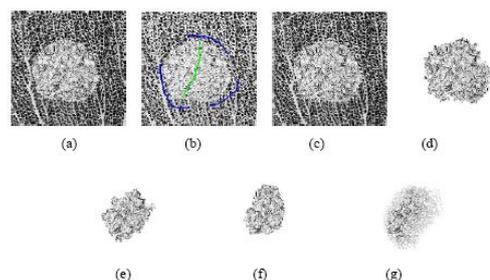

Fig. 10. The first comparison for texture image. (a) Texture 1. (b) User marker and SLIC segmentation. (c) Object boundary. (d) Proposed method. (e) RWR. (f) User-Friendly. (g) SLRW.
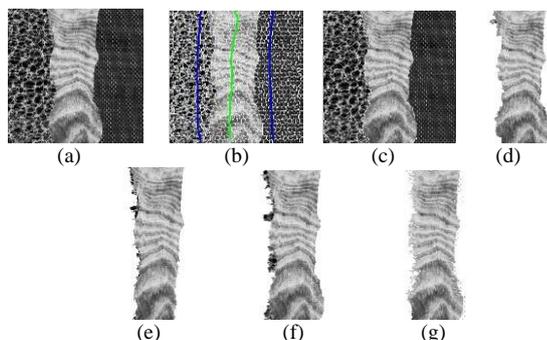
Fig. 11. The second comparison for texture image. (a) Texture 2. (b) User marker and SLIC segmentation. (c) Object boundary. (d) Proposed method. (e) RWR. (f) User-Friendly. (g) SLRW.
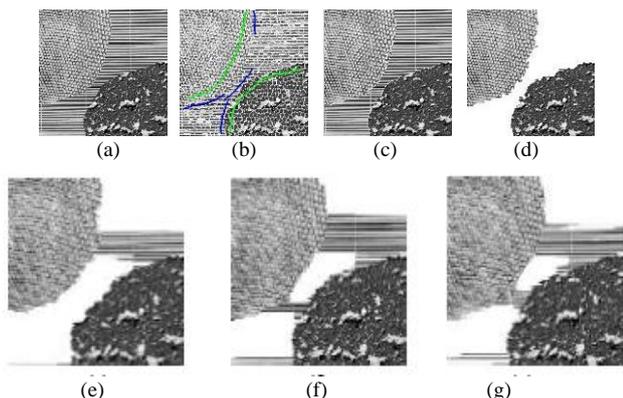

Fig. 12. The third comparison for texture image. (a) Texture 3. (b) User marker and SLIC segmentation. (c) Object boundary. (d) Proposed method. (e) RWR. (f) User-Friendly. (g) SLRW.

From Fig. 10 to Fig. 12, it is evident that our method provides the best result. These experimental results confirm the effectiveness to extract the objects of texture images. RWR, User-Friendly and SLRW hardly distinguish different textures. Especially in the third comparison, object almost cannot be extracted by these algorithms. On the contrary, the object is basically extracted by our algorithm.

V. CONCLUSIONS AND FUTURE DIRECTIONS

This paper proposes a novel interactive image segmentation based on color histogram and Contourlet transform. The image is initially segmented by SLIC segmentation and the users only need to provide the number of superpixels and mark the object by using some strokes. Then, a novel superpixel merging mechanism based on color histogram and Contourlet transform is proposed to represent each superpixel and merge objec and background superpixels individually according to the proposed superpixel feature. The proposed scheme efficiently exploits the color and frequency similarity of the target objects so that it is robust to the variations of input markers. Especially, it is also reliable to segment texture images.

In the future, we will continue to improve the performance of proposed method. First, it ought to be more adaptive. The number of superpixels must be provided at the beginning, but it does not make sense that we provide the same number for all kinds of images. Hence we need to figure out an adaptive method in accordance with the size or frequency distribution of images, which can provide us the number of superpixels. Second, the proposed method needs to be more robust and accurate to segment different kinds of images. Especially, the boundary of object superpixel extracted at the beginning is essential for the final segmentation result. Therefore, we may improve the SLIC segmentation or put forward a more accurate superpixel generation method.

REFERENCES

[1] Y. Y. Boykov and M. P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in *Proc. IEEE Int'l Conf. Computer Vision*, 2001, vol. 1, pp. 105-112.

[2] B. Peng, L. Zhang, and D. Zhang *et al.*, "Image segmentation by iterated region merging with localized graph cuts," *Pattern Recogn.*, vol. 44, no. 10-11, pp. 2527-2538, 2011.

[3] S. Ran, L. Zhi *et al.*, "Interactive Object Segmentation Using Iterative Adjustable Graph Cut," in *Proc. 2011 IEEE Visual Communications and Image Processing (VCIP)*, pp. 1-4.

[4] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut': Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309-314, 2004.

[5] B. L. Price, B. Morse, and S. Cohen, "Geodesic Graph Cut for Interactive Image Segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3161-3168.

[6] E. Moschidis and J. Graham, "A Systematic Performance Evaluation of Interactive Image Segmentation Methods Based on Simulated User Interaction," in *Proc. IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2010, pp. 928-931.

[7] L. Grady, "Random walks for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768-1783, 2006.

[8] T. H. Kim, K. M. Lee, and S. U. Lee, "Generative smage segmentation using random walks with restart," in *Proc. the 10th European Conference on Computer Vision: Part III*, 2008, pp. 264-275.

[9] Y. Yi, Y. Gao, and W. Li *et al.*, "Improved random walker interactive image segmentation algorithm for texture image segmentation," in *Proc. Chinese Control and Decision Conference (CCDC)*, 2011, pp. 4163-4166.

[10] W. Ning, H. Lin-Lin, and Z. Baochang, "A Fast Hybrid Method for Interactive Liver Segmentation," in *Proc. 2010 Chinese Conference on Pattern Recognition (CCPR)*, pp. 1-5.

[11] L. Yugang and Y. Yizhou, "Interactive image segmentation based on level sets of probabilities," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 2, pp. 202-213, 2012.

[12] B. C. McCary and Y. Cao, "Probabilistic interactive segmentation using multiphase level sets," in *Proc. Southwest10*, pp. 141-144.

[13] S. Cheng, A. Goel *et al.*, "Interactive segmentation of medical images using belief propagation with level sets," in *Proc. IEEE International Conference on Image Processing*, 2010, pp. 4113-4116.

[14] Z. Lei and J. Qiang, "A Bayesian Network Model for Automatic and Interactive Image Segmentation," *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2582-2593, 2011.

[15] M. Haimin, "Research on Interactive Segmentation Algorithm Based on Search Path Optimization," in *Proc. International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2009, pp. 286-289.

[16] M. A. E. Saban and B. S. Manjunath, "Interactive segmentation using curve evolution and relevance feedback," in *Proc. IEEE International Conference on Image Processing*, 2004, pp. 2725-2728.

[17] D. Lei, A. Yilmaz, and Y. Rong, "Interactive image segmentation using Dirichlet process multiple-view learning," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2119-2129, 2012.

[18] B. Yang, Q.-Y. Zhao, R. Zhang *et al.*, "Receptive Field Based Image Modeling Method for Interactive Segmentation," in *Proc. 2nd International Congress on Image and Signal Processing*, pp. 1-4.

[19] T. V. Spina, J. A. Montoya-Zegarra, A. X. Falcao *et al.*, "Fast interactive segmentation of natural images using the image foresting

transform," in *Proc. 16th International Conference on Digital Signal Processing*, 2009, pp. 1-8.

[20] A. Protiere and G. Sapiro, "Interactive image segmentation via adaptive weighted distances," *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 1046-1057, 2007.

[21] A. A. Hussein and Y. Xiaochun, "A statistical approach to interactive image segmentation," in *Proc. International Conference on Multimedia Technology*, 2011, pp. 5260-5263.

[22] T. Chunna and G. Xinbo, "Fast interactive segmentation algorithm of image sequences based on relative fuzzy connectedness," *Journal of Systems Engineering and Electronics*, vol. 16, no. 4, pp. 750-755, 2005.

[23] J. F. Ning, L. Zhang, D. Zhang *et al.*, "Interactive image segmentation by maximal similarity based region merging," *Pattern Recognition*, vol. 43, no. 2, pp. 445-456, Feb. 2010.

[24] R. Achanta, A. Shaji, K. Smith *et al.*, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274-2282, 2012.

[25] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091-2106, 2005.

[26] Y. Wenxian, C. Jianfei, Z. Jianmin *et al.*, "User-friendly interactive image segmentation through unified combinatorial user inputs," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2470-2479, 2010.

[27] X. Shiming, P. Chunhong, N. Feiping *et al.*, "Interactive image segmentation with multiple linear reconstructions in windows," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 342-352, 2011.

Guoheng Huang received his B.Sc. degree in applied mathematics and M.E degree in software engineering from South China Normal University. He is currently a Ph.D. student majoring in software engineering at the University of Macau. His research interests include image/video processing and pattern recognition.

Chi-Man Pun received the B.Sc. and M.Sc. degrees from the University of Macau, Taipa, Macau, in 1995 and 1998, respectively, and the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, Shatin, Hong Kong, in 2002. He currently is an associate professor at the Department of Computer and Information Science, University of Macau. His research interests include digital watermarking, multimedia compression, analysis and processing, pattern recognition, and computer vision.