International Journal of Multimedia and Ubiquitous Engineering
Vol. 10, No. 1 (2015), pp. 159-172
http://dx.doi.org/10.14257/ijmue.2015.10.1.15

Automatic Frame Composition Using Histogram Based Graph Cut

Daehee Kim, Hyungtae Kim, Jinho Park, Donggyun Kim and Joonki Paik

Chung-Ang University, Seoul, Korea
wangcho100@gmail.com, paikj@cau.ac.kr

Abstract

In this paper, we present an automatic background composition method using histogram-based graph cut. The proposed method consists of four steps: i) initial label map generation, ii) label map update, iii) object extraction by segmentation, and iv) dynamic background composition. Since the proposed method can minimize the user interaction for generating the initial label map and updating, it is suitable for simple interaction using a low-speed processor and limited memory space. Experimental results show that the proposed method provides better segmentation results compared with existing state-of-the-art methods with significantly reduced computational complexity. The proposed automatic object segmentation and background composition method can be applied to video editing, video conference, and video contents creation using low-cost mobile devices such as smart phones, smart TVs, and tablet PCs.

Keywords: *Histogram based graph cut, object segmentation, trimap*

1. Introduction

Traditional image segmentation aims to extract homogeneous regions with respect to texture or color properties, while recent segmentation methods can be defined as a process which typically partitions an image into meaningful objects according to some pre-specified semantics [1]. However, it is still difficult to locate the desired object based on a unified detection method due to various object features: color, intensity, shape, and contour, to name a few. In order to achieve better segmentation performance, many interactive methods have been proposed [2-5], which require the user interaction to manually define the desired object in advance [1].

There were to solve the problem of inconvenient user-interaction. Boykov proposed an optimization-based graph cut algorithm using a very simple user interaction [2]. Li proposed a system for cutting out a moving object from a video clip and pasting it onto another video or background image using three-dimensional (3D) graph-cut [3]. To achieve this goal, the user is required to select a few key frames in the video sequence at every ten frames for separating foreground and background using the snapping tool [4]. Wang proposed an interactive system for extracting foreground objects from a video that allows users to easily indicate the foreground objects across space and time [5]. Ning proposed an interactive object segmentation algorithm by using maximum similarity-based region merging (MSRM) [6]. Kim proposed an automatic object segmentation algorithm by using histogram-based graph cut (HBGC) and label map generation [7]. Since both HBGC and MSRM based methods require interactive initial label map, it is necessary to automatically generate the initial label map for low-cost, automated applications.

In this paper, we present a novel automatic background composition and label map generation. Under assumption that a moving object generates the target region to be segmented, the proposed algorithm consists of four steps: i) initial label map generation,

ii) label map update, iii) object extraction by segmentation, and iv) background composition as shown in Figure 1.

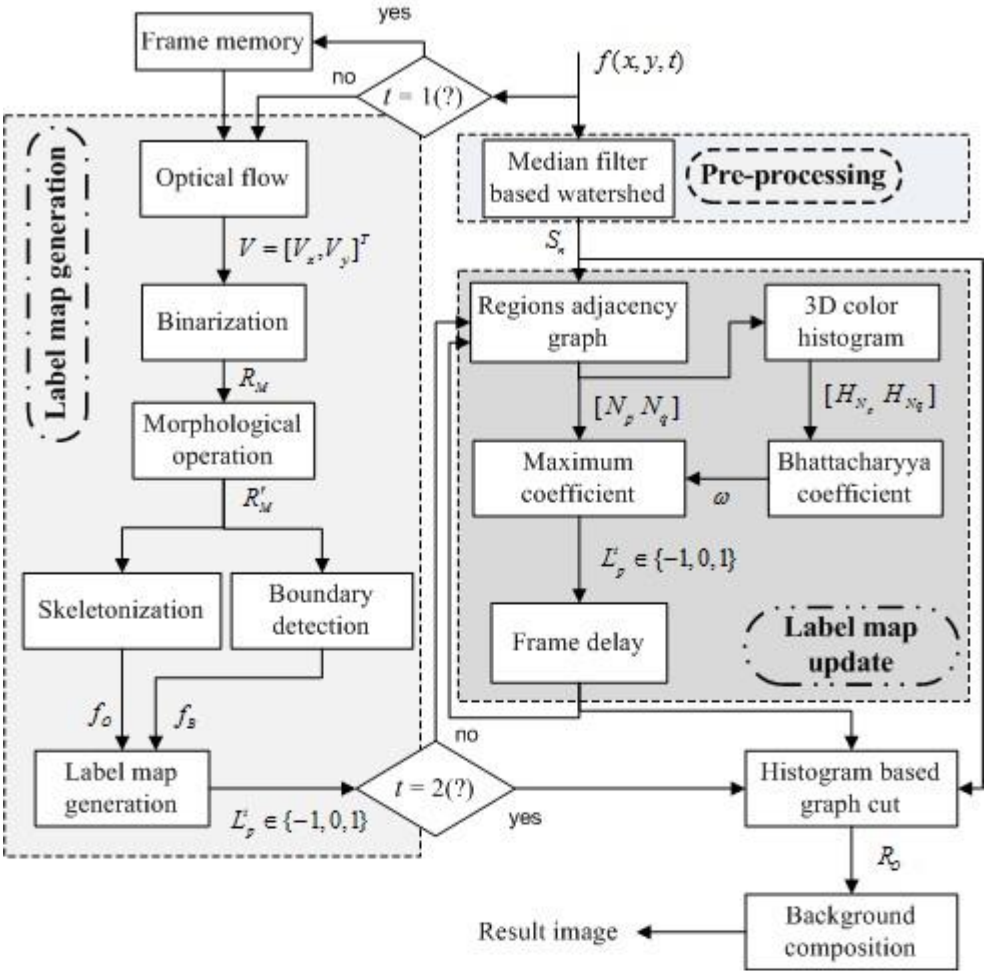


Figure 1. The Flowchart of the Proposed Object Segmentation and Background Composition Algorithm

More specifically, the input image is first split into sufficiently many regions using the median filter-based watershed algorithm [8]. The initial label map is then generated by using optical-flow, skeletonization [9], and morphological boundary detection. The label map is updated by using color histogram and Bhattacharyya coefficients for the following frame. The segmentation result is obtained using HBGC. We then compose the image frame using the segmented object and a priori stored background. At this point, the motion data in the background is measured by the optical-flow, and the final result is synthesized by moving the object region according to the motion of the background.

Although the primary object of the proposed method is to secure the privacy of individual users using a low-cost video editing device, it can also be used to create high-quality video contents using seamless composition of an object and background.

This paper is organized as follows. Section 2 presents the automatic object segmentation algorithm, and section 3 presents the background composition algorithm. Section 4 summarizes experimental results, and Section 5 concludes the paper.

2. Automatic Object Segmentation Method

Most existing object segmentation methods use one or combination of color-based merging, mean-shift or motion-based segmentation, graph-cut, and grow-cut algorithms, which are not suitable for video object segmentation in their original forms particularly in applications of consumer low-cost devices with limited computational power. In order to fit the consumer environment, we use automatic initial label map generation and its update for detecting an object without user interaction, and then apply the HBGC algorithm.

For improving the segmentation speed of the low-performance mobile processor, we first over-segment the input image. The proposed over-segmentation method uses the median filter-based watershed algorithm to generate N nodes, S_n, $i = 1, \cdots, N$, which are the output of the median filter-based watershed block shown in Figure 1 [7].

2.1. Automatic Generation of the Initial Label Map

The proposed label map generation process is shown in the left dotted block denoted “Label map generation” as shown in Figure 1. Based on the assumption that a moving region is an object of interest in the video, we generate an initial label map using optical-flow. Let $V = [V_x, V_y]^T$ be the velocity vector, and then the optical flow equation is expressed as

$$\left\langle \nabla f(x,y,t), v(x,y,t) \right\rangle + \frac{\partial f(x,y,t)}{\partial t} = 0, \tag{1}$$

where (x, y) represents the image coordinate, and t the time index, and $\langle \cdot, \cdot \rangle$ denotes the vector inner product. The moving region, denoted by R_M, can be defined as

$$R_M = \left\{ (x,y) \mid \sqrt{V_x^2 + V_y^2} > \theta_M \right\}, \tag{2}$$

where θ_M is a pre-specified threshold. We empirically used $\theta_M = 0.35$ for the visually best segmentation result. We refine the detected moving region R_M using morphological opening and closing operations by removing noise and filling holes. The refined region, denoted by R'_M, fuses narrow breaks and long thin gulfs to eliminate small holes and fill gaps on the contour.

Skeletonization 0 is applied to R'_M for stable object region labeling. Let $f_O(x, y)$ be the result of skeletonization. If $f_O(x, y)$ is true, the node that contains (x, y) has the label $L_p = 1$. Morphological operation is applied to R'_M for stable background region labeling. Let $f_B(x, y)$ be the boundary of results of the morphological dilation operation. If $f_B(x, y)$ is true, the node that contains (x, y) has the label $L_p = -1$. The initial label map generation process can be expressed as

$$L_p(x,y) = \begin{cases} 1, & f_O(x,y,t) = \text{true} \\ -1, & f_B(x,y,t) = \text{true}. \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Figure 2 shows the result of label map generation. In Figure 2(f), the white curve represents the background region labeling, while the black curve represents the object region labeling.

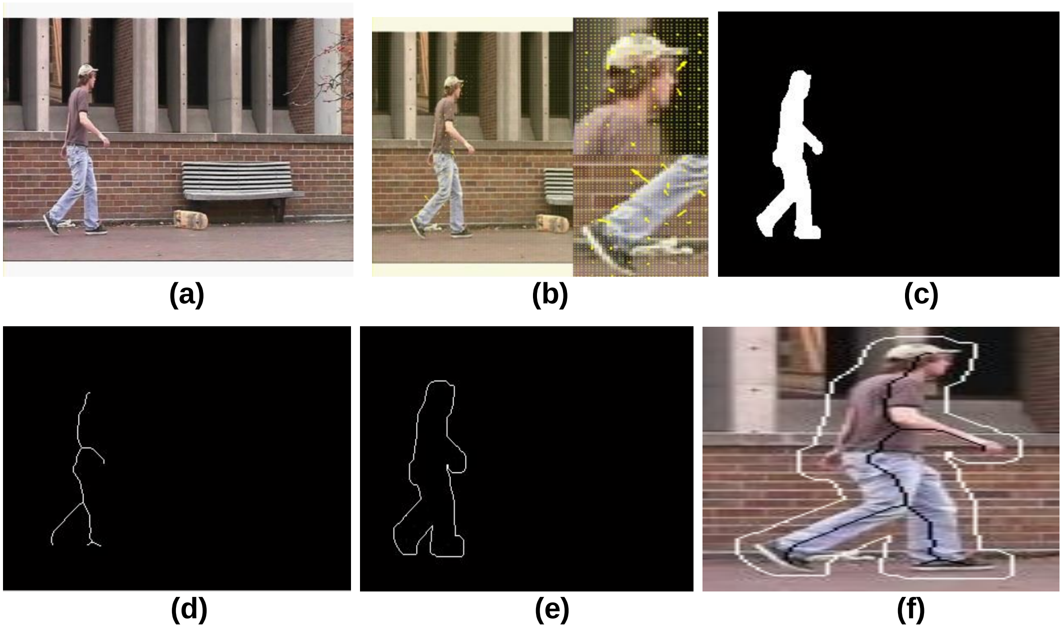


Figure 2. Result of Label Map Generation: (a) Input Image, (b) the Estimated Optical Flow, (c) the Refined Motion Region, (d) the Result of Skeletonization, (e) Detected Boundary, and (f) the Generated Label Map

2.2. Label Map Update

In static video, object's motion is limited. We can thus estimate the label map of the current frame $f(x,y,t)$ from that of the previous frame $f(x,y,t-1)$, which yields stable segmentation with reduced computational load. The estimation error is corrected by updating the label map based on the over- segmentation result S_n.

For updating the label map in consecutive frames, we first detect adjacent nodes using the regions adjacency graph (RAG) at the $t+1$ st frame, and then select the most similar node using color histogram and Bhattacharyya coefficient. Let N_p be the node containing the label map L_p in the previous frame, then the corresponding adjacent node in the current frame, denoted by N_q, is obtained by using the RAG.

Trémeau et al. claimed that the most similar node can be obtained as [10]

$$G(N_p, N_q) = \min_{s_i \in N_1, s_j \in N_2, (s_i, s_j) \in E} \lambda((s_i, s_j)), \tag{4}$$

where $G=(S,E)$ is an undirected graph, $s_i \in S$ is a set of regions corresponding to image elements, E is a set of edges connecting pairs of neighboring regions. Each edge, $(s_i, s_j) \in E$, has the corresponding weight $\lambda(s_i, s_j)$ that represents the dissimilarity of the two regions connected by that edge. A region is represented by a component $N \subseteq S$. We obtain the dissimilarity between two neighboring regions $N_1, N_2 \subseteq S$ as the minimum weighted edge connecting them.

Figure 3 shows the graph structure of a partitioning example and it's RAG.

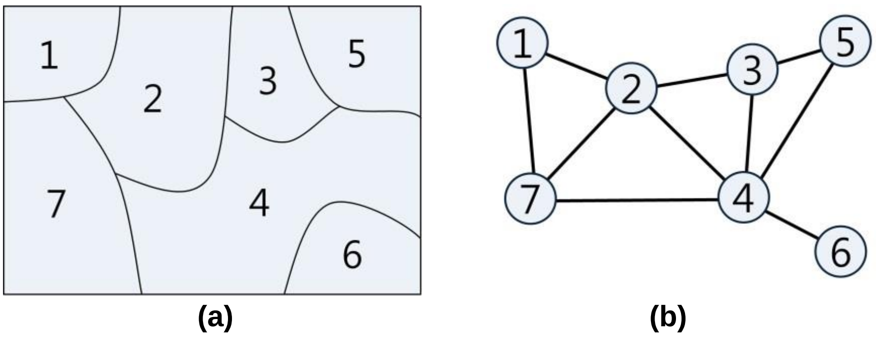


Figure 3. An Example of (a) Partitioned Regions and (b) the Corresponding RAG

RAG is used to calculate similarity with the surrounding region in the N_p of current frame. And N_p can be calculated with based on obtained in the L_p of the previous frame. N_p and N_q in the current frame are shown in Figure 4.

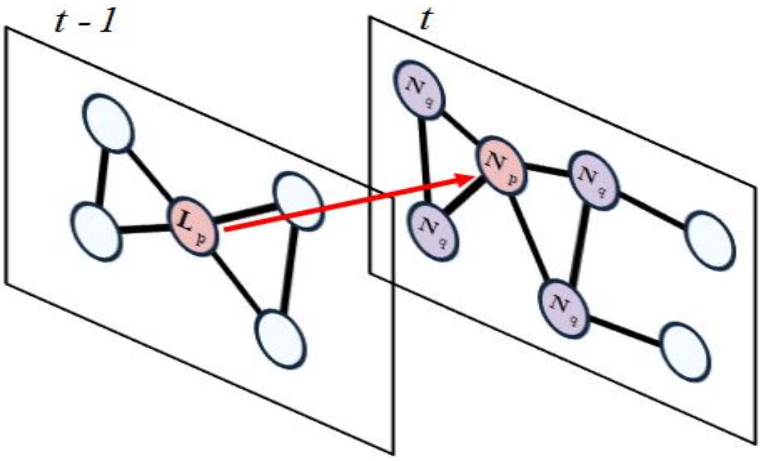


Figure 4. An Updating Process of the Initial Label Map

We uniformly quantize each color channel into 16 levels and then the histogram of each region is calculated in the feature space of $16 \times 16 \times 16 = 4096$ bins. We use the Bhattacharyya coefficient ω to measure the similarity between H_{N_p} and H_{N_q} as

$$\omega = \sum_{n=1}^{4096} \sqrt{H_{N_p}^{n}, H_{N_q}^{n}}, \tag{5}$$

where H_{N_p} and H_{N_q} are the normalized histograms of N_p and N_q, respectively, and the superscript n represents the n-th element of the corresponding histogram. The geometric explanation of the Bhattacharyya coefficient actually reflects the perceptual similarity between regions. If two regions have similar contents, their histograms will also be similar, and hence their Bhattacharyya coefficients will increase [6]. We will therefore update L_p by the node N_q of maximum ω.

2.3. HBGC for Object Segmentation

Computational efficiency and user-friendly way of operation are critical issues in video editing environment and mobile devices. So we use the HBGC algorithm, which is a modified version of the original graph-cut algorithm, for reducing computational load and minimizing user interactions. The existing graph cut algorithm minimizes the Gibbs energy expressed as

$$E(L) = \lambda \sum_{p \in I} R(p, L_p) + \sum_{\{p,q\} \in N} B(p_i, q_j)\delta(L_p \neq L_q), \tag{6}$$

where

$$\delta(L_p \neq L_q) = \begin{cases} 1, & L_p \neq L_q \\ 0, & L_p = L_q \end{cases}, \tag{7}$$

The positive regularization parameter λ specifies a relative importance of the region properties term $R(p, L_p)$ versus the boundary properties term $B(p_i, q_j)$ [2]. $R(p, L_p)$ encodes the color similarity of a node, indicating if it belongs to the foreground or background as

$$R(p, L_p) = -\text{In}\,\Pr(I_p \mid L_p), \tag{8}$$

where I_p is the 3D feature vector of pixels of the node p in the RGB color space, and $\Pr(I_p \mid L_p)$ is the conditional probability of the I_p given the L_p. So $R(p, L_p)$ reflects the likelihood that p belongs to L_p, and thereby can be used to penalize the incorrect label assignment.

Since the existing algorithm is computationally expensive, we present a novel object, efficient segmentation method by comparing histograms of the corresponding regions. More specifically, we redefine $R(p, L_p)$ by using color histograms and Bhattacharyya coefficients as

$$R(p, L_p) = \min \sum_{n=1}^{4096} \sqrt{H_p^n, H_{(L_p=1, L_p=-1)}^n}, \tag{9}$$

where H_p and $H_{(L_p=1, L_p=-1)}$ represent normalized histograms of p and L_p, respectively. Then, the boundary term $B(p_i, q_j)$ is given as

$$B(p_i, q_j) = \left| p_i - q_j \right| \cdot g(C_{ij}), \tag{10}$$

where $g(\xi) = 1/(\xi + 1)$, and $C_{ij} = \left\| C(i) - C(j) \right\|^2$ is the l_2 -Norm of the RGB color difference of two pixels i and j. $C(i)$ and $C(j)$ represent the color of i and j, respectively. Note that $\left| p_i - q_j \right|$ allows us to capture the gradient information along the segmentation boundary. In other words, $B(p_i, q_i)$ plays a role of the penalty term when adjacent nodes are assigned with different labels. The more similar the colors of the two nodes are, the larger $B(p_i, q_i)$ becomes, and thus the less likely the edge is on the object boundary. The globally optimum solution for the minimization of the energy function can be found by using the min-cut/max-flow algorithm [11].

3. Background Composition

For natural replacement of background while keeping the foreground object, we can use blue screen and Trimap-based algorithms. A blue screen is a special device for color-based extraction of the foreground object. The Trimap-based composition algorithm requires additional user interaction, which is needed for high-quality, accurate composition [12-14].

The background composition algorithm used in this work starts with a new background image to be replaced with the existing one. In case of video, the background may change if the camera moves. For solving this problem we propose a novel background composition algorithm for natural images as shown in Figure 5.

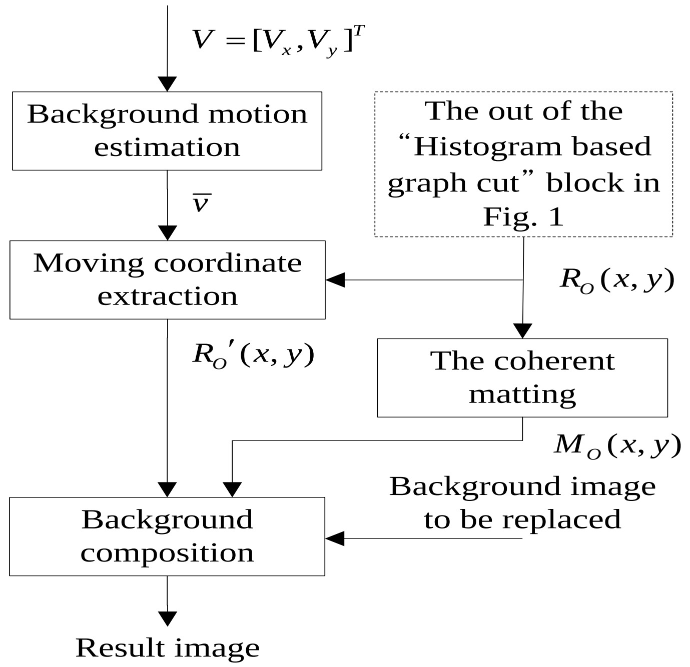

$V = [V_x, V_y]^T$
Background motion estimation
The out of the “Histogram based graph cut” block in Fig. 1
$\bar{v}$
Moving coordinate extraction
$R_O(x, y)$
$R_O'(x, y)$
The coherent matting
$M_O(x, y)$
Background image to be replaced
Background composition
Result image

Figure 5. The Proposed Background Composition Algorithm

After estimating the background motion in the video, the location of composition is reset by using the estimated value. For reducing unnatural artifacts, we apply the coherent matting algorithm [15] in the boundary of the foreground.

3.1. Estimation of Background Motion

If a consumer hand-held camera system is used, motions in the background result in erroneously generated background images. In this work, we estimate the background motion in a specific region to overcome the camera shaking problem. Since an object of interest is, in general, present in the center of the frame, we assign two regions in upper left and right corners of the image for estimating the background motion.

The object coordinate moves along the average of motions estimated in the two regions v_1 and v_2 as

$$\bar{v} = \frac{1}{2W}\left(\sum v_1(V_x, V_y) + \sum v_2(V_x, V_y)\right) , \tag{11}$$

where W represents the window size. We experimentally used a 16×16 window for background motion estimation. The object coordinate $R_O(x, y)$ is updated by using the $\bar{v}$ as

$$R_O'\begin{bmatrix} x \\ y \end{bmatrix} = R_O\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \bar{v}_x \\ \bar{v}_y \end{bmatrix} . \tag{12}$$

The updated object coordinate $R_O'(x, y)$ is combined with a new background image as shown Figure 6.

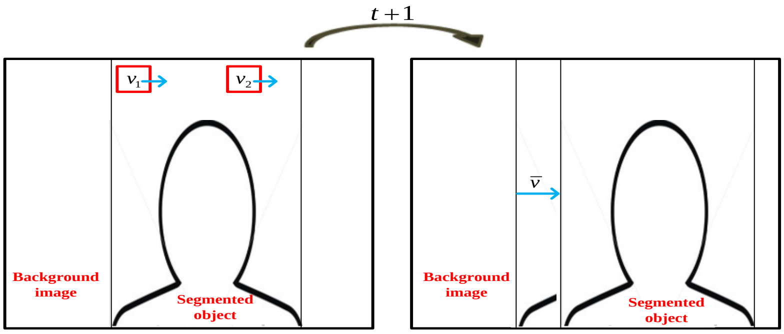


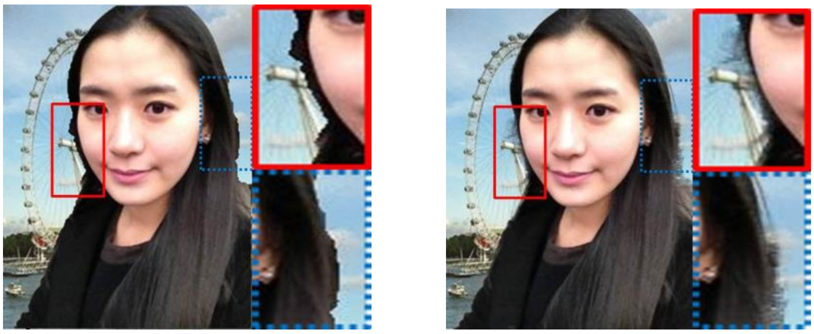
Figure 6. The Combined Coordinate of the Updated Object

3.2. Refinement of Composite Boundary

Synthetic combination of the foreground and background images may produce an unnatural-looking boundary. Chuang has proposed a natural composition method using the Trimap in the complex background [12, 14], which needs a user interaction.

For making the composition result look more natural, we applied the coherent matting algorithm [15] in the boundary region between the object and background. The coherent matting algorithm improves Bayesian matting by introducing a regularization term called α . The generated alpha matte complies with the prior binary segmentation boundaries, and performs better than Bayesian matting when foreground and background colors are similar.

Uncertain regions in matting are computed by dilating the binary object boundary, typically by 7 pixels. For small holes or thin gaps in the foreground, this dilation may result in no background colors to be sampled nearby. In this case, we instead sample background colors from neighboring frames. Figure 7 shows the result of refined boundary of an object.



(a) (b)

Figure 7. Comparing the Results of Different Background Composition Methods: (a) Without Refined Boundary and (b) Using the Coherent Matting Method

4. Experimental Results

In order to evaluate performance of the proposed algorithm, we used several test videos, which were taken by a commercial mobile phone camera. We also tested a set of still images. The sizes of foreground test frames are 320×480, and those of background images are 640×480.

Figure 8 and Table 1 compare performance of the proposed and two existing object segmentation algorithms [2], [6] in the sense of both subjective and objective manners, respectively. While existing algorithms need user interaction to specify the initial label map, the proposed method can automatically generate the initial label map.

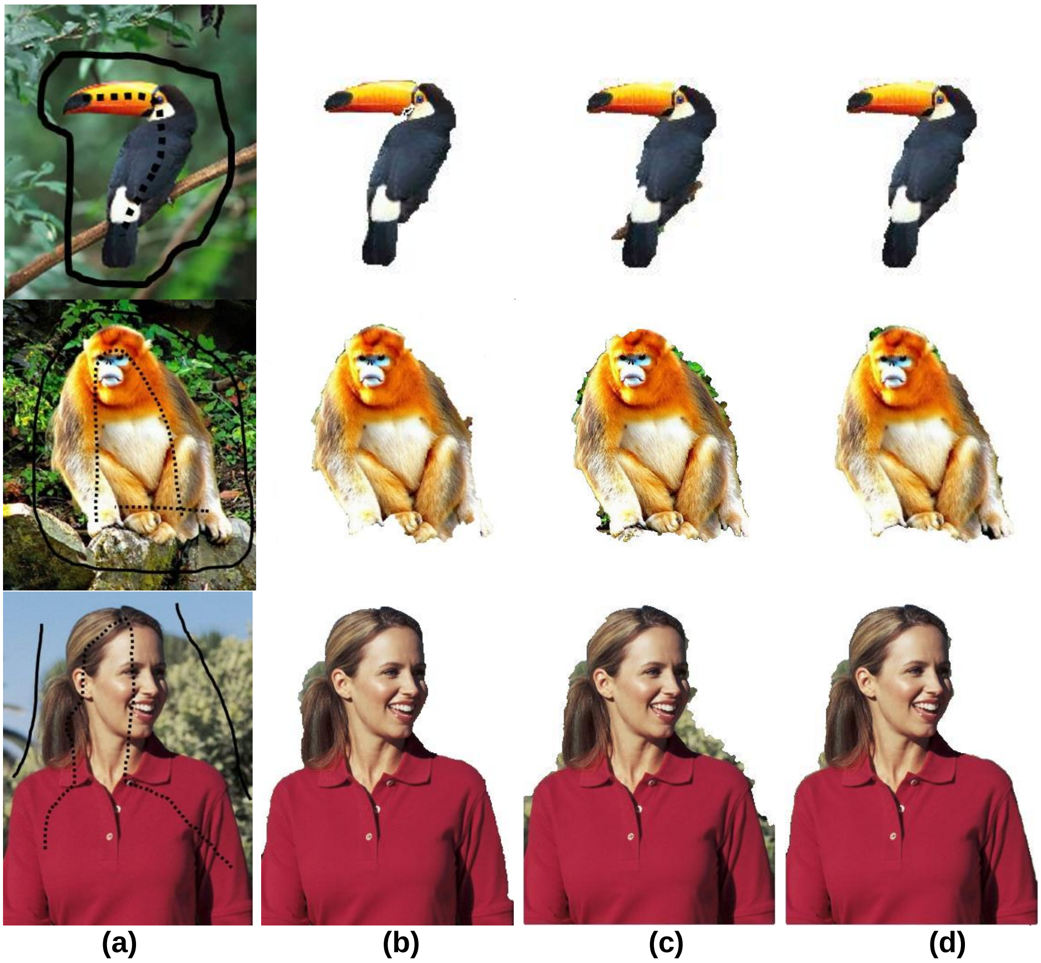


Figure 8. Results of Object Segmentation using Three Different Algorithms: (a) iInput Images with the Initial Label Map (solid line: background, dotted line: object), (b) Segmentation Results of Boykov's Method [2], (c) Segmentation Results of Ning's method [6], (d) Segmentation Results of the Proposed Method

Table 1. Processing Times (in seconds) of Three Different Segmentation Algorithms using a Dual Core CPU at 2.50 GHz

Image	Boykov [2]	Ning [6]	Proposed
Bird	0.089	45.792	0.053
Monkey	0.110	60.120	0.056
Woman	0.095	51.356	0.051

Figure 9 shows the step-by-step results of the proposed segmentation and composition algorithm in still images. The size of the test image is 320×480 .

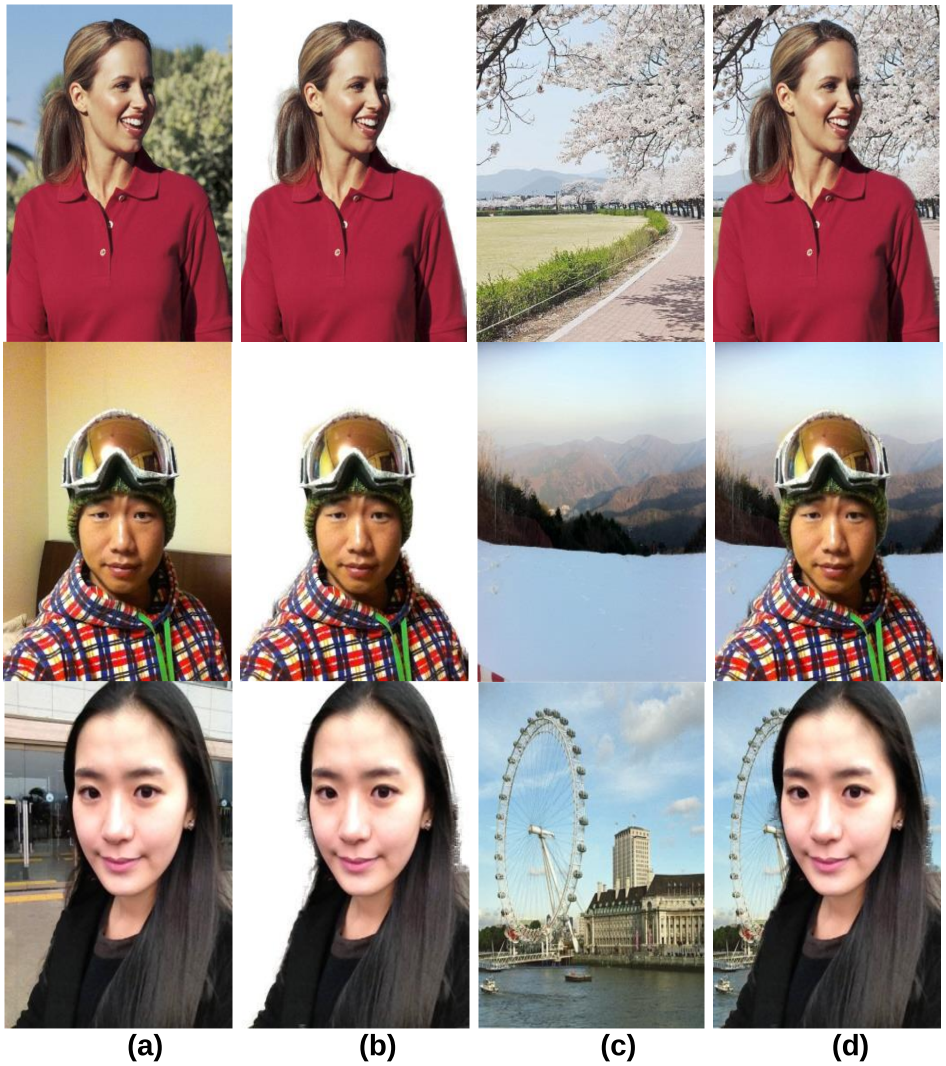


Figure 9. Step-by-step Results of the Proposed Segmentation Algorithm: (a) Input Image, (b) the Segmented Object Using the Coherent Matting Method, (c) the New Background Image, and (d) the Combination of the Segmented Object and the New Background.

Figure 10 shows the composition results using dynamic background. The size of object image is 320×480 , and the size of background image is 640×480 . The processing time was 0.051 seconds/frame, which can make real-time implementation possible. Since the latest smart phones adopt a dual or quad core CPU at 1.4GHz or higher, the proposed method can provide the similar performance and processing time in real mobile devices.

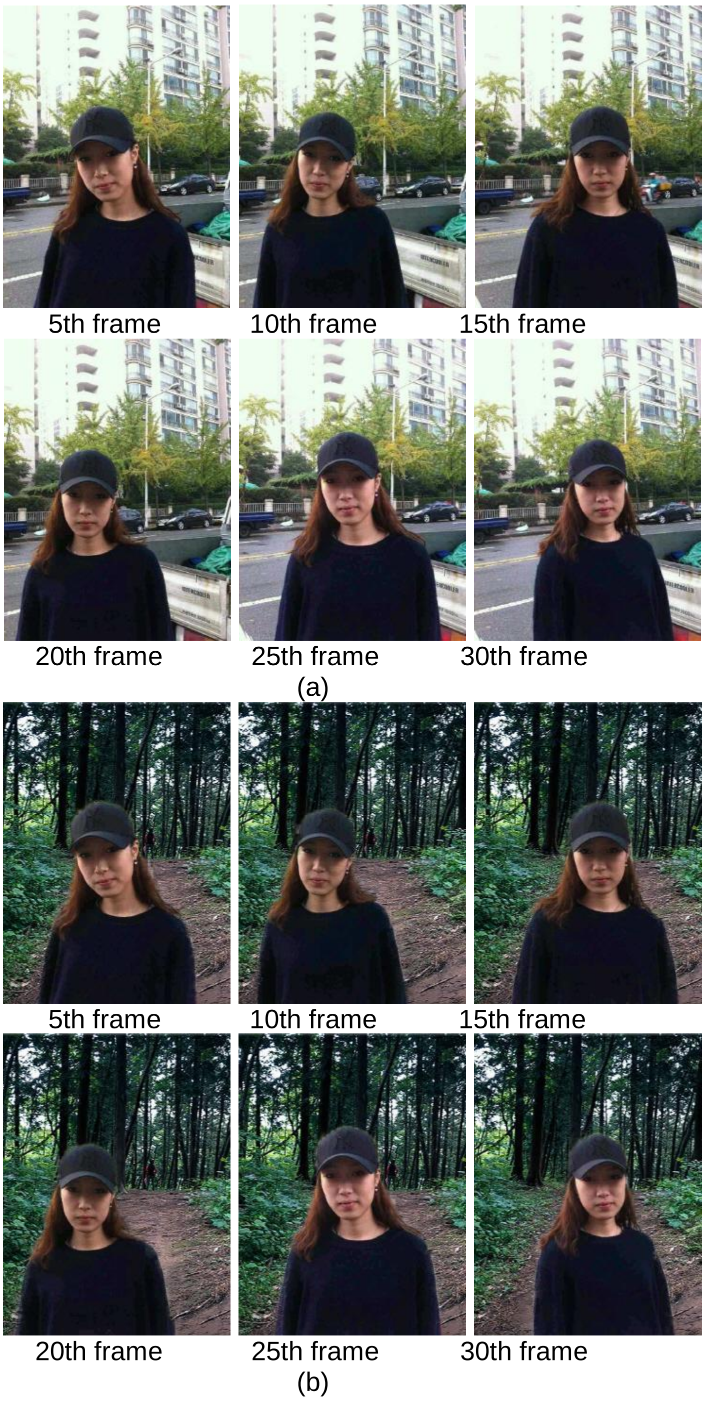


Figure 10. The Results of Composition Using Dynamic Background; (a) Input Image and (b) the Result of Background Composition

5. Conclusion

An automatic object segmentation and background composition method has been presented for interactive video matting. For automatic segmentation of moving objects in video, the initial label map is generated and updated without user interaction. The HBGC improves the performance of existing graph cut-based algorithms. Dynamic background is correctly synthesized by estimating motion information and compensating the coordinate of the moving object. The proposed method can be applied to the wide range of consumer video services including; video editing, video calls, and contents creation especially in commercial low-cost mobile devices.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2013R1A1A2060470) and by Seoul R&BD Program (PA130170).

References

[1] H. Li, K. Ngan and Q. Liu, “FaceSeg: Automatic Face Segmentation for Real-Time Video”, IEEE Trans. Multimedia, vol. 11, no. 1, **(2009)**.

[2] Y. Boykov and M. Jolly, “Graph Cuts and Efficient N-D Image Segmentation”, International Journal of Computer Vision, vol. 70, no. 2, **(2006)**.

[3] Y. Li, J. Sun and H. Shum, “Video Object Cut and Paste”, ACM Trans. on Graphics, vol. 24, **(2005)**.

[4] Y. Li, J. Sun, C. Tang and H. Shum, “Lazy Snapping”, ACM Trans. on Graphics, vol. 23, **(2004)**.

[5] J. Wang, P. Bhat, R. Colburn, M. Agrawala and M. Cohen, “Interactive Video Cutout”, ACM Trans. on Graphics, vol. 24, **(2005)**.

[6] J. Ning, L. Zhang, D. Zhang and C. Wu, “Interactive Image Segmentation by Maximal Similarity Based Region Merging”, Pattern Recognition, vol. 43, **(2010)**.

[7] D. Kim and J. Paik, “Automatic Moving Object Segmentation Using Histogram-Based Graph Cut and Label Maps”, Electronics Letters, vol. 48, no. 19, **(2012)**.

[8] L. Vincent and P. Soille, “Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, **(1991)**.

[9] Y. Wan, L. Yao, B. Xu and P. Zeng, “A Distance Map Based Skeletonization Algorithm and Its Application in Fiber Recognition”, IEEE Int. Conf. Audio, Language and Image Processing, **(2008)**; Shanghai, China.

[10] A. Trémeau and P. Colantoni, “Regions Adjacency Graph Applied to Color Image Segmentation”, IEEE Trans. Image Processing, vol. 9, **(2000)**.

[11] Y. Boykov and V. Kolmogorov, “An Experimental Comparison of Min-cut/Max-Flow Algorithms for Energy Minimization in Vision”, IEEE Trans. Pattern Anal, Mach. Intell., vol. 26, no. 9, **(2004)**.

[12] J. Wang and M. Cohen, “Image and Video Matting: A Survey”, Computer Graphics and Vision, vol. 3, no. 2, **(2007)**.

[13] Y. Chuang, A. Agarwala, B. Curless, D. Salesin and R. Szeliski, “Video Matting of Complex Scenes”, ACM Trans. on Graphics, vol. 21, **(2002)**.

[14] Y. Chuang, B. Curless, D. Salesin and R. Szeliski, “A Bayesian Approach to Digital Matting”, IEEE Conf. Computer Vision and Pattern Recognition, **(2001)**; Hawaii, USA.

[15] H. Shum, J. Sun, S. Yamazaki, Y. Li and C. Tang, “Pop-up light field: An Interactive Image-based Modeling and Rendering System”, ACM Trans. on Graphics, vol. 23, **(2004)**.

Authors

Daehee Kim was born in Buchun, Korea in 1981. He received the B.S. degree in computer engineering from Kangnam University, Korea, in 2005. He received his M.S. and the Ph.D. degrees in image engineering from Chung-Ang University, Korea, in 2007 and 2013, respectively. Currently, he is a research fellow professor in Chung-Ang University, Korea.

Hyungtae Kim was born in Seoul, Korea in 1986. He received his B.S. degree in the department of electrical engineering from Suwon University in 2012. He is currently pursuing a M.S. degree in image processing at Chung-Ang University.

Jinho Park was born in Seoul, Korea in 1987. He received the B.S. degree in electronic engineering from Suwon University, Korea, in 2013. Currently, he is pursuing the M.S. degree in image processing at Chung-Ang University.

Donggyun Kim was born in Busan, Korea in 1983. He received B.S. and M.S. degrees in electronic and electrical engineering from Chung-Ang University, Korea, in 2007 and 2009, respectively. Currently, he is pursuing the Ph.D. degree in image processing at Chung-Ang University.

Joonki Paik was born in Seoul, Korea in 1960. He received the B.S. degree in control and instrumentation engineering from Seoul National University in 1984. He received the M.S. and the Ph.D. degrees in electrical engineering and computer science from Northwestern University in 1987 and 1990, respectively. From 1990 to 1993, he joined Samsung Electronics, where he designed the image stabilization chip sets for consumer's camcorders. Since 1993, he has joined the faculty at Chung-Ang University, Seoul, Korea, where he is currently a Professor in the Graduate school of Advanced Imaging Science, Multimedia and Film.