Reprinted from
Journal of Imaging Science and Technology® 60(6): 060409-1–060409-7, 2016.

Adaptive Combination of Local Motion, Appearance, and Shape for Video Segmentation

Woo-sung Shim
DMC R&D Center, Samsung Electronics Co., Ltd, Seoul, Korea

Se-hoon Kim
Mobile Communication Business, Samsung Electronics Co., Ltd, Suwon, Korea

Soochahn Lee
Department of Electronic Engineering, Soonchunhyang University, Asan, Korea
E-mail: sclsch@sch.ac.kr

Abstract. *In this paper, we propose an accurate and robust video segmentation method. The main contributions are threefold: (1) multiple cues (appearance and shape) are explicitly used and adaptively combined to determine segment probability; (2) motion is implicitly used to compute the shape cue; and (3) the segment labeling is improved by utilizing geodesic graph cuts. Experimental results show the effectiveness of the proposed method. © 2016 Society for Imaging Science and Technology.*

INTRODUCTION

Video segmentation is important for video manipulation tasks such as composition, compression, and 2D–3D conversion, as well as image understanding tasks such as object recognition. Considerable research has been done especially for bilayer segmentation of video, where a single foreground (FG) object is segmented from the background (BG).[1–3] The recent state-of-the-art methods utilize pixel-wise probabilistic information based on the local appearance and shape propagated from the previous segmented frame using motion.[2]

In this paper, we propose an accurate and robust video segmentation method. The main contributions are threefold: (1) multiple cues (appearance and shape) are explicitly used and adaptively combined to determine segment probability; (2) motion is implicitly used for the shape cue; (3) segment labeling is improved by incorporating geodesics, i.e., using geodesic graph cuts[4] rather than simple graph cuts.[3] Experimental evaluation demonstrates the effectiveness of the proposed method.

RELATED WORK

To solve image and video segmentation problems, many researchers have used either texture or edge information.[5–7] For example, the Magic Wand method[6,7] starts with a user-specified point or region to compute a region of connected pixels such that all the selected pixels fall within some adjustable tolerance of the color statistics of the specified region. While the user interface is straightforward, finding the correct tolerance level is often cumbersome and sometimes impossible. The Intelligent Scissors method[5] allows a user to choose a “minimum cost contour” by roughly tracing the object’s boundary with the mouse. As the mouse moves, the minimum cost path from the cursor position back to the last “seed” point is shown. If the computed path deviates from the desired one, additional user-specified “seed” points are necessary.

While easier than just selecting pixels manually with a traditional selection tool, commercial image editing tools still demand a large amount of attention from the user. As a result, the user must control the curve carefully. If a mistake is made, the user has to “back up” the curve and try again. New methods have been developed for reducing human interactions.[8–10] Graph cut is a particularly powerful optimization technique that can be used for interactive segmentation.[3] This method works by allowing the user to give loose hints as to which parts of the image are foreground or background without enclosing regions or being pixel accurate. These hints usually take the form of clicking or dragging on foreground or background elements. Thus, it is very quick and easy to use for single images.

However, it may require human interaction for every frame in a video, which can accumulate to an excessive amount. Therefore, several methods specifically tailored for video based on graph cuts have been proposed.[2,13,14] Both the method by Wang et al.[13] and the method by Li et al.[14] treat the video as a 3D grid and apply graph cut. In this approach, the user inputs may be given in the 3D voxel space[13] or as image-based inputs for selected key frames.[14] The limitation of 3D grid based video segmentation is that the motions between successive frames are not sufficiently modeled. The *SnapCut* method by Bai et al.[2] uses a different approach, where each frame is segmented sequentially based on explicit motion estimation to infer the object boundary.

Received July 18, 2016; accepted for publication Oct. 18, 2016; published online Dec. 5, 2016. Associate Editor: Yeong-Ho Ha.

This approach tends to give more robust results for complex objects and environments.

Now, we briefly review the graph-cut segmentation method and the *SnapCut* method, which is closely related to the proposed method. The graph-cut segmentation method is based on optimization of the following cost function:[3]

$$\boldsymbol{E}(L) = \sum_{x_i \in P} R(x_i) + \lambda \cdot \sum_{x_i, x_j \in N_i} V(x_i, x_j), \qquad (1)$$

where $l_i \in \{F, B\}$ is the segmentation label (F and B represent foreground and background, respectively) for pixel $x_i \in \boldsymbol{P}$ and $R(x_i)$ is a region cost term based on the label $L(x_i) = l_i \in \{F, B\}$. $V(x_i, x_j)$ is a boundary cost between x_i and x_j, where N_i is the set of pixels around x_i. The parameter λ is a relative weight between the region cost R and the boundary cost V. Generally, the boundary cost corresponds to a measure of the similarity between the colors of adjacent pixels and the region cost is based on color models of the foreground and background. The graph-cut method minimizes Eq. (1) by casting the problem as a graph partitioning using the min-cut/max-flow graph algorithm.

The *SnapCut* method is summarized as follows: (1) the region cost term $R(x_i)$ in Eq. (1) is obtained by mixing local FG/BG probabilities for all $x_i \in \boldsymbol{P}$; (2) local FG/BG probability is obtained by adaptively combining color and shape probabilities. More details are as follows: Given the true segmentation of frame t, local windows are defined on the segmentation boundary, and the respective local appearance and shape information are propagated to frame $t+1$ by moving windows with motion vectors estimated by optical flow. Inside each local window, the respective appearance of FG and BG, modeled as Gaussian Mixture Models (GMMs), and the shape, motion compensated segmentation labels of frame t, are adaptively combined to compute local pixel-wise segment probabilities. Then, the local probabilities are mixed to construct global pixel-wise probability and the graph-cut method to assign pixel-wise labels from the segment probabilities.

Although the graph cut[3] is one of the most widely used methods for segmentation, it has a well-known shortcoming, called shrinking bias, causing bias toward shorter boundary length. Recently, the geodesic graph-cut method was proposed[4], where geodesics based on pixel segment probability[1] are incorporated into the graph-cut cost function. This method alleviates the problem of the shrinking bias considerably.

PROPOSED ALGORITHM

The proposed framework is based on the adaptive localized classifiers of the *SnapCut* method,[2] with several major improvements to enhance segmentation accuracy. Figure 1 shows an overview and Figure 2 summarizes each step of the proposed method.

Review of Adaptive Localized Classifiers

It is assumed that a ground-truth segment mask of the object on the first frame is available as an initial seed mask. From the initial mask, a number of overlapping local windows are created along the boundary of the FG object. Within each local window, color and shape statistics are computed for FG and BG segments, respectively, and applied as pixel-wise probability priors. Color is modeled by GMMs $\boldsymbol{G}_t = \{G_{F,t}, G_{B,t}\}$ while shape is modeled by the previous segment mask and a shape confidence map f_s.

For a pixel x, its color-based FG probability is defined as:

$$p_c(x) = p_c(x|F)/(p_c(x|F)+p_c(x|B)), \qquad (2)$$

where $p_c(x|F)$ and $p_c(x|B)$ are the corresponding probabilities computed from the two GMMs.

The color and shape prior weights are adaptively determined by the color confidence f_c, which measures how separable the local FG color is against that of the BG. With $L_t(x)$ as the segmentation mask of frame t, f_c is defined as:

$$f_c = 1 - \left(\int_{W_k} \left| L_t(x) - p_c(x) \right| \cdot \omega_c(x)\, \mathrm{d}x \right) \Big/ \int_{W_k} \omega_c(x)\ \mathrm{d}x, \qquad (3)$$

where W_k is the kth local window and $\omega_c(x)$ is weighting function $\omega_c(x) = exp(-d^2(x)/\sigma_c^2)$, where $d(x)$ is the spatial distance between x and the FG boundary, computed by the distance transform. Since $\omega_c(x)$ is larger for pixels

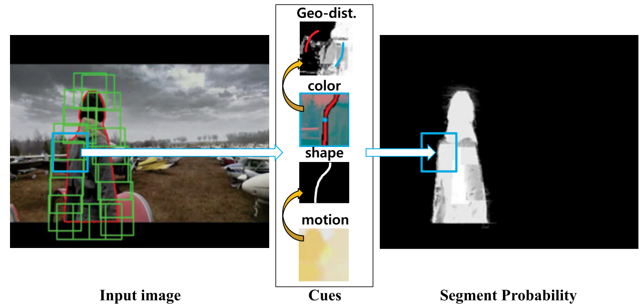


Figure 1. Overview of the proposed method. We combine image gradient based geodesics, color, shape and motion segmentation cues to accurately propagate object and background segment probabilities to adjacent video frames.

1. Initialize
Locate local windows & build models

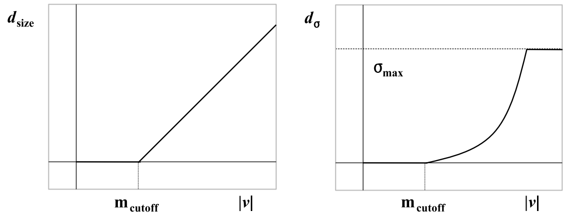
2. Local Window Update
Motion estimation & model update

3. Local Probability & Cost
Compute Shape-Color Probability
Compute Local Geodesic Cost

4. Adaptive Cost Integration
Local Probability & Cost Integration

5. Geodesic Graph Cut
Min-cut/Max-flow Optimization

Figure 2. The framework of the proposed method.



(a) $h(|v|)$ (b) $g(|v|)$

Figure 3. Motion cue mapping functions.

closer to the boundary, the color prior influence increases when the color-based probabilities agree well with $L_t(x)$, especially near the FG boundary. This is achieved by the shape confidence map $f_s(x) = 1 - exp(-(d^2(x)/\sigma_s^2))$, where the parameter σ_s is defined based on f_c as:

$$\sigma_s = \begin{cases} \sigma_{\min} + \sigma_d \cdot \left(\frac{f_c - f_{\text{cutoff}}}{1 - f_{\text{cutoff}}}\right)^r & f_{\text{cutoff}} < f_c \leq 1, \\ \sigma_{\min} & 0 \leq f_c \leq f_{\text{cutoff}}, \end{cases} \tag{4}$$

where $\sigma_d = \sigma_{\max} - \sigma_{\min}$. We use $f_{\text{cutoff}} = 0.85$, $\sigma_{\min} = 2$, $r = 2$, and $\sigma_{\max}$ equals the size of the local window. It is clear that a high f_c will result in a large σ_s, resulting in a loose shape constraint, and vice versa. Finally, the FG probability $p_{cs}^k(x)$ is defined as:

$$p_{cs}(x) = f_s(x) \cdot L_{t+1}(x) + (1 - f_s(x))p_c(x). \tag{5}$$

Local classifiers are propagated to adjacent frames based on motion estimation. Specifically, each local window is moved based on a two-step motion estimation stage comprising (1) global affine motion estimation based on matched speeded-up robust feature points[15] and (2) pixel-wise optical flow. The displacement vector for a window is defined as the average motion of pixels within that window. For the newly positioned windows of the next frame, the shape prior, i.e., the segment mask is updated based on the estimated motion, and the color prior is augmented with GMMs built from the new frame. Among the GMMs from the previous frame and the new frame, the one that gives the larger FG region is selected.

Implicit Motion Cue

In a fast moving region, when the motion is large, it is observed that the local motion estimation (optical flow) gives a relatively large error, and a shape deformation occurs. Under this observation, we propose a method to improve segmentation probability by utilizing the local motion information.

As created in the initialization step, the local window size is fixed. When a large local motion estimation error occurs from a relatively large motion of the object, the local window may not fully cover the local region at $t + 1$ corresponding to a region at t. Although a large local window alleviates this misalignment problem, it causes an inaccurate motion vector result due to averaging over a large area. Hence, instead of using a large window by default, we dynamically adjust the window size during the window propagation step (see the local model update in Ref. 3) in accordance with the motion magnitude as follows:

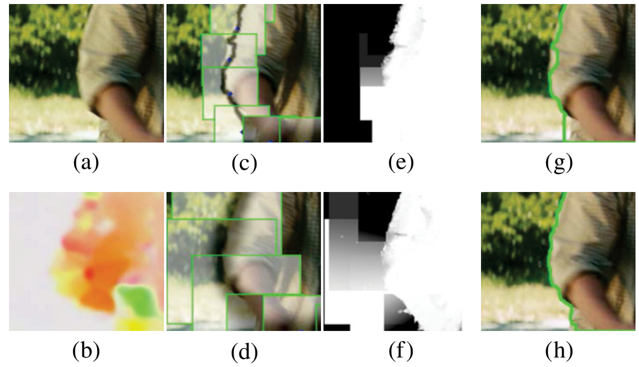


Figure 4. Improving foreground probability using implicit motion cue: (a) input frame, (b) optical flow, (c) and (d) local windows and shape confidence, (e) and (f) corresponding foreground probability, and (g) and (h) resulting segment without/with implicit motion cue, respectively.

$$s^k(t+1) = s_{\text{default}} + h(|\bar{v}^k|), \tag{6}$$

where $s^k(t + 1)$ is the new size of the ith window at time $t + 1$, s_{default} is the default window size, $|\bar{v}^k| = \sqrt{\bar{v}_{x,k}^2 + \bar{v}_{y,k}^2}$ is the magnitude of averaged optical flow $(v_{x,k}, v_{y,k})$ for the kth window and $h(\cdot)$ is a mapping function of the motion to window size.

As described above, we assume that shape changes more in fast moving regions than small motion regions. Under this assumption, the magnitude of a motion would be a factor to decide the shape confidence $f_s(x)$, as well as the color confidence f_c. Moreover, the motion-based shape confidence can reduce the side effect of the motion-based window size adjustment in Eq. (6): including shape with long contour pixels increases a possibility of foreground probability $p_{cs}(x)$ to be carved where the shape is mismatched due to motion estimation error and rapid shape change (see Figure 4). As the result, we decide to use the motion magnitude for adjusting the shape confidence $f_s(x)$. The shape confidence $f_s(x)$ is affected by σ_s in Eq. (4), which is determined by f_c and other parameters. If motion magnitude is directly used to determine σ_s, the color confidence information is eliminated from the shape confidence f_s. Instead, we use the motion to modify the $\sigma_{\min}$ as follows:

$$\sigma_{\min}^k = \bar{\sigma}_{\min} + g(|\bar{\boldsymbol{v}}^k|), \tag{7}$$

where $\bar{\sigma}_{\min}$ is the default value of $\sigma_{\min}$. Then we compute σ_s in Eq. (4), for the kth window, using $\sigma_{\min}^k$ in Eq. (7) as $\sigma_{\min}$ in Eq. (4). The profiles of $h(|\bar{v}|)$ in Eq. (6) and $g(|\bar{v}|)$ in Eq. (7) are shown in Figures 3(a) and 3(b), respectively.

Fig. 4 shows the effectiveness of the motion-cue-based window size and shape confidence compensation. The overlaid gray regions, in Fig. 4(c) and (d), represent a shape confidence map $f_s(x)$. The motion estimation error and shape change incur an incorrect and very narrow shape confidence map (high confidence on the shape) so that the resulting ambiguity in the foreground probabilities yields incorrect segmentation. On the other hand, the proposed method alleviates the problem, giving a more accurate segmentation result.

Local Geodesics

As mentioned in the overview, we utilize the geodesic graph-cut algorithm to alleviate the shrinking bias and enhance robustness to motion estimation error. However, geodesic graph cut is an interactive image segmentation method that uses the geodesic distance from a user's scribble to a pixel. The method cannot be easily integrated into our non-interactive and local window based video segmentation framework.

In this section, we attempt to develop a method to automatically create geodesic seeds, i.e., without user interaction, for each local window. Precise creation of geodesic seeds is critical for achieving good foreground and background geodesic costs because the geodesic distance is very sensitive to seed placement.

It is obvious that foreground/background seeds should be placed in correct regions where they are separated well. If we assume that each local window is propagated well and the local region is covered by the window, we can utilize the already existing good information, namely, color and shape probabilities. From this observation, we devise a simple method to construct local seeds using the color–shape probability $p_{cs}(x)$ in (5) as follows:

$$\begin{aligned}\Omega_F &= \{x; p_{cs}(x) \geq T_{seed}\},\\ \Omega_B &= \{x;\, 1 - p_{cs}(x) \geq T_{seed}\},\end{aligned} \tag{8}$$

where the threshold $T_{seed} = 0.95$ is used for our experiment. The upper part of Figure 5(c) shows an example of created geodesic seeds, where white pixels represent foreground seeds and black pixels represent background seeds. The local geodesic distance for each window is computed as

$$D_l(x) = \min_{s \in \Omega_l} d_l(s, x), \tag{9}$$

where Ω_l is the set of points (seeds) with label $l \in \{F, B\}$. The geodesic distance from a point to another point according to the color model for the label l is given by

$$d_l(a, b) = \min_{L_{a,b}} \int_0^1 |\nabla p_l(s) \cdot \dot{L}_{a,b}(s)|\, ds, \tag{10}$$

where $L_{a,b}$ is a path from a to b parameterized by $s = [0, 1]$ and $\nabla p_l(s) = \nabla p_c(L_{a,b}(s)|l)$ is the gradient of color probability $p_c(x|l)$ along the path $L_{a,b}$. Finally the local geodesic cost is obtained as follows:

$$G_l(x) = D_l(x)/(D_F(x) + D_B(x)). \tag{11}$$

Fig. 5 shows the effectiveness of combining geodesic cost with the color–shape probability. The wrong shape confidence results in wrong segmentation (see Fig. 5(a) and (b)). The foreground cost is rather improved (dotted red ellipse in Fig. 5(d)) by combining the geodesic cost using the seeds generated as described above (Fig. 5(c)).

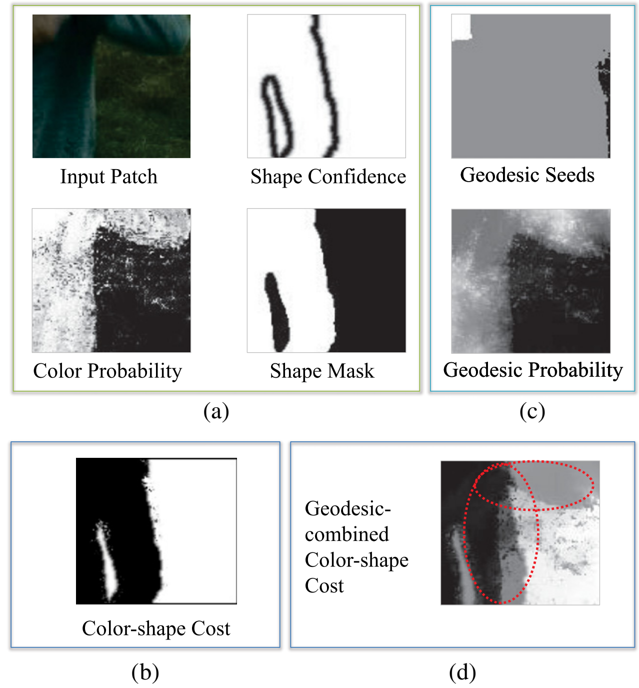


Figure 5. Improving segmentation probability by combining geodesics.

Geodesic Graph-cut Segmentation

Now, we need to obtain the region cost E_R and the boundary cost E_B in Eq. (1) to segment foreground out of background by the graph-cut method via min-cut/max-flow optimization algorithm.

First, we combine color–shape probability $p_{cs}(x)$ in Eq. (5) and geodesic distance $G_l(x)$ in (11) to make the local region cost for each local window as follows:

$$\begin{aligned}R_F^k(x) &= (1 - p_{cs}^k(x)) + u(x) \cdot G_F^k(x),\\ R_B^k(x) &= p_{cs}^k(x) + u(x) \cdot G_B^k(x),\end{aligned} \tag{12}$$

where $\boldsymbol{k}$ is the index of local windows and the local geodesic weight function $u(x)$ is the confidence of geodesics computed by

$$u(x) = \left| \frac{D_F(x) - D_B(x)}{D_F(x) + D_B(x)} \right|^{2.5}. \tag{13}$$

Note that, to convert the probability $p_{cs}^k(x)$ to a cost, $1-p_{cs}^k(x)$, rather than $p_{cs}^k(x)$, is combined into $R_F^k(x)$.

Then we integrate all local region costs $R_l^k(x)$ in Eq. (12) into a global region cost $R_l(x)$. Since the windows overlap each other, we obtain the global region cost $R_l(x)$ by weighted combination of the local region costs as

$$R_l(x) = \sum_k (R_l^k(x) \cdot \omega_k(x)) / \sum_k \omega_k(x), \tag{14}$$

where the weight function $\omega_k(x) = exp(-(x - c_k)^2/\sigma^2)$.

For the boundary cost term V, we compute the boundary cost at a pixel x by measuring color difference with

(a) without window resizing (b) with window resizing

Figure 6. A sample result for comparison between with and without motion-based local window resizing.

eight neighboring pixels as follows:

$$b(x, x_i)_{x_i \in N} = \frac{1}{d_p(x, x_i)} \cdot e^{-\{\beta \cdot d_c(c(x)-c(x_i))^2\}}, \quad (15)$$

where $d_p(x_i,x_j)$ and $d_c(c(x_i)-c(x_j))$ are position difference and color difference between pixels x_i and x_j, respectively. The parameter β is computed as given in Ref. 1. Then, we add the boundary cost weight corresponding to $u(x)$ in Eq. (13), and to resolve the jagged boundary problem, we also combine the global probability of boundary (gPb)[16] as follows:

$$V(x, x_i) = (1 + \bar{u}(x, x_i) + gPb(x)) \cdot b(x, x_i), \quad (16)$$

where geodesics-based boundary weight $\bar{u}(x, x_i) = (u(x) + u(x_i))/2$, and $gPb(x)$ is the probability of boundary at pixel x.

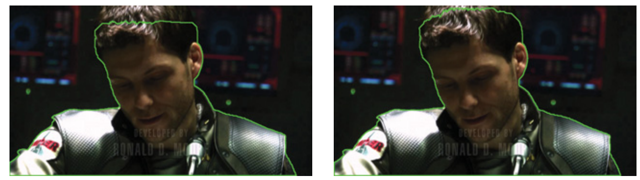
Finally, we can segment, i.e., obtain pixel-wise label, foreground and background by minimizing cost function (1) with Eqs. (14) and (16). The resulting segment mask $L_{t+1}(x)$ is used for segmenting the next frame at time $t+1$.

Other Refinements

Although a full-frame foreground probability map can be constructed in the way described in previous sections, and they can give good results in many cases, this process can be iterated to generate more accurate segments. This iterative refinement is effective, especially in cases where a large motion occurs. The iteration scheme is simple: repeat the process using $L_{t+1}(x)$ as an initial mask.

Another refinement we must consider is when the local window is located on the frame border. For such a local window, the segmentation of a target object will occur along the frame border line since there is no background beyond the frame border. This causes a problem if the object moves inward in the next frame such that the segmented boundary of the border line becomes a shape prior for the next frame. We resolve this problem by examining the history of the current local windows if they were on the frame border in the previous frame. If it is determined that the current local window was on the frame border previously, the shape prior is assumed false and the local color confidence level is dynamically increased. This forces the system to ignore the segmented border line and look for a new segmentation boundary.

(a) input frame #3 (b) matched features with previous frame



(c) a result without boundary compensation (d) a result with boundary compensation

Figure 7. A sample result showing effectiveness of boundary compensation.

RESULTS AND DISCUSSION

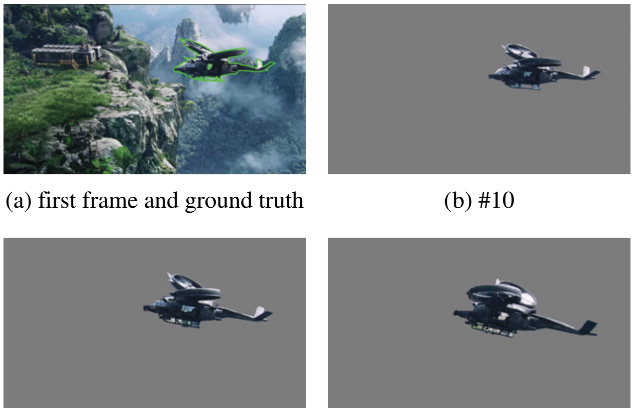
We present experimental results on a variety of image sequences from movies and TV.

Figure 6 presents improved segmentation performance by using motion-based window resizing described in the Implicit Motion Cue section. In the test scene, Harry raises his arm to the face rapidly while the torso is almost static. When fixed windows are used, portions of the figure's hand and head are missed due to over-fitted appearance modeling and insufficient compensation of motion estimation error. These problems are overcome by increased local window size based on the motion estimation.

Figure 7 shows the effectiveness of the proposed method to deal with the new appearance on the image boundary. As shown in Fig. 7(b), the upper hair unseen in the previous frame appears due to camera movement (upper direction). Without shape confidence compensation on the frame border, the image boundary on the previous frame is preserved as an object's boundary as in Fig. 7(c). Figure 7(d) shows that, although there are segmentation errors at the corner of the head, the newly appeared hair is segmented relatively well.

Figure 8 presents the results of segmentation of the proposed method for the Avatar sequence. In the scene, the foreground and background are of low contrast in color; the scene is rather cluttered by the trees, rocks, and grasses. Although, some background is labeled as foreground in the landing gear region of the helicopter, the proposed method resulted in a reasonable segmentation, considering the complexity of foreground and background scenes.

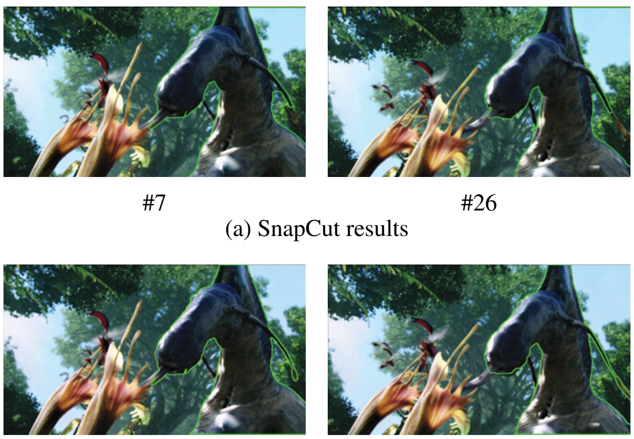
We compare the proposed method to *SnapCut*,[3] which has been transferred into the *Roto Brush* released in Adobe After Effects CS5. As shown in Figure 9, the thin and long horns of an alien horse are preserved in the proposed method.



(a) first frame and ground truth (b) #10

(c) #28 (d) #49

Figure 8. Segmentation results for Avatar-1 sequence.



#7 #26

(a) SnapCut results

#7 #26

(b) our results

Figure 9. A result comparing *SnapCut* to ours for Avatar-2 sequence.

We further provide a quantitative comparison with more recent semi-supervised video segmentation methods that are all based on an initial or a sparse number of user annotated frames.[18–21] Specifically, we evaluated the proposed method using the benchmark dataset recently made public by Perazzi et al.,[17] and compared the results with other semi-supervised methods, as presented in Table I. Here, the region similarity J is the Jaccard index, which is defined as the intersection over-union of the estimated segmentation and the ground-truth mask. The contour accuracy F is defined as $F = (2P_cR_c/P_c + R_c)$, the F-measure of P_c and R_c, which are the precision and recall of the segmentation boundary, respectively. In order to be robust to small inaccuracies, contour-based precision and recall P_c and R_c are computed via a bipartite graph matching.[17] Finally, J/F-mean denotes the mean for all sequences, while J/F-object recall measures the fraction of sequences scoring higher than a threshold 0.5.

While the performance of the proposed method was better than the method by Chang et al. (temporal superpixels [TSP]),[18] it was worse than the methods by Ramakanth and Babu SeamSeg (SEA),[19] Grundmann et al. Hierarchical graph-based video segmentation (HVS),[20] and Fan et al.

Table I. Quantitative comparison of region similarity (J) and contour accuracy (F).

Measure J-mean J-object recall				
TSP Ref. 18	SEA Ref. 19	HVS Ref. 20	JMP Ref.21	Proposed
0.358	0.556	0.596	0.607	0.451
0.388	0.606	0.698	0.693	0.482
F-mean F-object recall				
0.346	0.533	0.576	0.586	0.455
0.329	0.559	0.712	0.656	0.459

JumpCut (JMP).[21] While the temporal relationship between only a pair of frames is analyzed in the proposed method, all the comparison methods focus on utilizing the appearance similarity and motion relationships of several frames at once. This seems to be the main reason for the difference in performance.

CONCLUSIONS

We proposed a novel video segmentation method that adaptively combines motion, appearance, and shape cues to compute the segment probability and adaptively determines the size of the local window according to video characteristics. Experimental results have shown that the proposed method is superior to the state-of-the-art segmentation method in cases where a large motion occurs, FG/BG color contrast is low, and the foreground object is complicated.

We believe that the proposed method can be applied to 2D–3D conversion of film and TV. Although recently, new movies are being filmed in 3D from the beginning, many existing 2D legacy footages and contents require conversion to 3D. The current conversion process involves precise manual operation, consisting of careful segmentation of individual objects using rotoscoping. To date, no computational solution exists that could replace the manual procedures in high-quality production. This incurs a heavy cost in conversion, ranging up to $100,000 per minute of converted footage. Our proposed method can be used to considerably cut down the manual work and the production cost.

For future work, we plan to extend the proposed method to further utilize appearance and motion information of multiple frames simultaneously. We believe that this extension may increase the robustness of the proposed method.

ACKNOWLEDGMENT

This work was supported by the Soonchunhyang University Research Fund.

REFERENCES

1 X. Bai and G. Sapiro, "A geodesic framework for fast interactive image and video segmentation and matting," J. Int. Computer Vis. **82**, 113–132 (2009).

2 X. Bai, J. Wang, D. Simons, and G. Sapiro, "Video SnapCut: Robust video object cutout using localized classifiers," *ACM Transactions on Graphics (TOG)* (2009), Vol. 28, p. 70.

3 Y. Boykov and G. Funka-Lea, "Graph cuts and efficient N-D image segmentation," Int. J. Comput. Vis. **70**, 109–131 (2006).

4 B. L. Price, B. S. Morse, and S. Cohen, "Geodesic graph cut for interactive image segmentation," *IEEE Conf. on Computer Vision and Pattern Recognition 2010* (IEEE, Piscataway, NJ, 2010), pp. 3161–3168.

5 E. Mortensen and W. Barrett, "Tobogan-based intelligent scissors with a four parameter edge model," *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR '99)* (IEEE, Piscataway, NJ, 1999), pp. 23–25.

6 Adobe Systems Incorp., In: Adobe Photoshop User Guide, 2002.

7 L. J. Reese and W. A. Barrett, "Image editing with intelligent paint," *The Annual Conf. European Association for Computer Graphics (Eurographics)* (Saarbruecken, Germany, 2002).

8 Y. Li, J. Sun, C. Tang, and H. Shum, "Lazy snapping," *ACM Transactions on Graphics (TOG) - Proc. ACM SIGGRAPH (2004)* (2004), Vol. 23, pp. 303–308.

9 C. Rother, V. Kolmogorov, and A. Blake, "'Grab Cut'—Interactive foreground extraction using iterated graph cuts," *ACM Tran. Graphics (TOG) — Proc. ACM SIGGRAPH (2004)* (2004), Vol. 23, pp. 309–314.

10 Y. Boykov and M. P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images," *IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR 2001)* (IEEE, Piscataway, NJ, 2001).

11 S. Khan and M. Shah, "Object based segmentation of video using color, motion and spatial information," *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR 2001)* (IEEE, Piscataway, NJ, 2001).

12 P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik, "Occlusion boundary detection and figure/ground assignment from optical flow," *IEEE Int'l. Conf. on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Piscataway, NJ, 2011).

13 J. Wang, P. Bhat, A. Colburn, M. Agrawala, and M. Cohen, "Interactive video cutout," *ACM Trans. Graphics (TOG) - Proceedings of ACM SIGGRAPH (2004)* (2004), pp. 585–594.

14 Y. Li, J. Sun, and H.-Y. Shum, "Video object cut and paste," *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2005* (2005), Vol. 24.

15 H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," Comput. Vis. Image Underst. **220**, 346–359 (2008).

16 P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," IEEE Trans. Pattern Anal. Mach. Intell. **33**, 898–916 (2011).

17 F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," *IEEE Computer Vision and Pattern Recognition (CVPR)* (IEEE, Piscataway, NJ, 2016).

18 J. Chang, D. Wei, and J. W. Fisher III, "A video representation using temporal superpixels," *IEEE Computer Vision and Pattern Recognition Conf. on Computer Vision (CVPR)* (IEEE, Piscataway, NJ, 2013).

19 S. A. Ramakanth and R. V. Babu, "SeamSeg: Video object segmentation using patch seams," *Proc. 2014 IEEE Conf. Computer Vision and Pattern Recognition (CVPR '14)* (IEEE, Piscataway, NJ, 2014), pp. 376–383, DOI:http://dx.doi.org/10.1109/CVPR.2014.55.

20 M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," *IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2010* (IEEE, Piscataway, NJ), pp. 2141–2148.

21 Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen, "Jumpcut: Non-successive mask transfer and interpolation for video cutout," *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH Asia 2015* (2015), Vol. 34, p. 195.