

## LETTER

## Virtual Halo Effect Using Graph-Cut Based Video Segmentation\*

Sungchan OH<sup>†a)</sup>, Hyug-Jae LEE<sup>†b)</sup>, Nonmembers, and Gyeonghwan KIM<sup>†c)</sup>, Member

**SUMMARY** This letter presents a method of adding a virtual halo effect to an object of interest in video sequences. A modified graph-cut segmentation algorithm extracts object layers. The halo is modeled by the accumulation of gradually changing Gaussians. With a synthesized blooming effect, the experimental results show that the proposed method conveys realistic halo effect.

**key words:** video segmentation, video editing, graph-cut, virtual halo effect

## 1. Introduction

The recent popularization of portable digital video devices has drawn attention to video editing methods, such as insertion of subtitles, rearranging video scenes and manipulation of object layers. For an easy access to video object layering for non-professionals, we introduce a scheme for video segmentation with simple user interaction only for the first frame of given video sequences. The effectiveness of the proposed video segmentation method is demonstrated by *virtual halo effects* which place synthetic bright light between background and object layers to highlight objects of interest.

An overview of the proposed system is shown in Fig. 1. Given a video sequence, grouping of pixels into the foreground and the background is done by the proposed sequential video segmentation algorithm based on graph-cut. The method constructs a spatio-temporal pixel-wise graph with two consecutive frames. Temporal connections in the graph are made from pixel correspondences based on motion. The motion contributes to video segmentation by propagating the previous results of segmentation and extracting moving objects as the foreground layers. To alleviate the problem of error propagation, the segmentation results from the previous frame are selectively used in the current segmentation process.

A halo model is constructed by a mixture of Gaussians and its variation over time is modeled by introducing random parameters and estimated motion vectors. To prevent unnatural transition between object layers and to realize plausible halo-overlaid video sequences, a simplified

model of blooming effect in CCD/CMOS image sensors is employed as well.

## 2. Video Segmentation Based on Graph-Cut

Typical energy function of well-known graph-cut [1], [2] consists of a data term for a node and a smoothness term for an edge between nodes as Eq. (1). In the proposed method, the data term,  $E_{data}$ , measures the preference of a node  $p$  for label  $L_p$ . The smoothness term,  $E_{smooth}$ , on the other hand, encourages neighboring nodes with similar color and motion to be assigned the same label.

$$E = \sum_{p \in \mathcal{V}} E_{data}(L_p) + \sum_{(p,q) \in \mathcal{E}} E_{smooth}(L_p, L_q) \quad (1)$$

Minimization of Eq. (1) is performed by  $\alpha$ -expansion algorithm [3]. Specific details of modification in graph-cut are explained in subsequent sections.

### 2.1 Spatio-Temporal Graph Construction

In the proposed method, a spatio-temporal graph  $G = \langle \mathcal{V}, \mathcal{E} \rangle$  is constructed with two consecutive frames,  $I(t-1)$  and  $I(t)$ . The set  $\mathcal{V}$  is composed of the previous frame node set  $\mathcal{V}_{t-1}$  and the current frame node set  $\mathcal{V}_t$ , where each node corresponds to each pixel in  $I(t-1)$  and  $I(t)$ . The edge set  $\mathcal{E}$  connects inter- and intra-frame nodes. While intra-frame edges connect 4-neighbors of intra-frame nodes as usual graph-cuts do, inter-frame edges link corresponding inter-frame nodes based on motion  $\mathbf{V}(t-1)$  [4] over  $I(t-1)$  and  $I(t)$  as shown in Fig. 2. Pink and blue nodes represent  $\mathcal{V}_{t-1}$  and  $\mathcal{V}_t$ , respectively. We designate  $L_{FG1}$  as an object of interest to be highlighted among other object labels in a set  $L_{FG} = \{L_{FG1}, L_{FG2}, \dots\}$ , and the background is regarded as one layer  $L_{BG}$ .

### 2.2 Data Cost

As the labeling process refers to the previous result, the user-provided seed information for the first frame propagates through video sequences. The label  $L^*$ , which is determined by the segmentation result of the  $(t-1)$ -th frame, is included in the data cost function. Since false segmentation is usually observed around boundary between differently moving objects, confidence of  $L^*$  needs to be diminished as the distance to the boundary decreases.

Manuscript received June 24, 2013.

Manuscript revised July 26, 2013.

<sup>†</sup>The authors are with the Department of Electronic Engineering, Sogang University, Republic of Korea.

\*This work was supported in part by Samsung Electronics Co. Ltd. and Sogang University Research Grant.

a) E-mail: amuckrunner@gmail.com

b) E-mail: miuri@sogang.ac.kr

c) E-mail: gkim@sogang.ac.kr (Corresponding author)

DOI: 10.1587/transinf.E96.D.2492

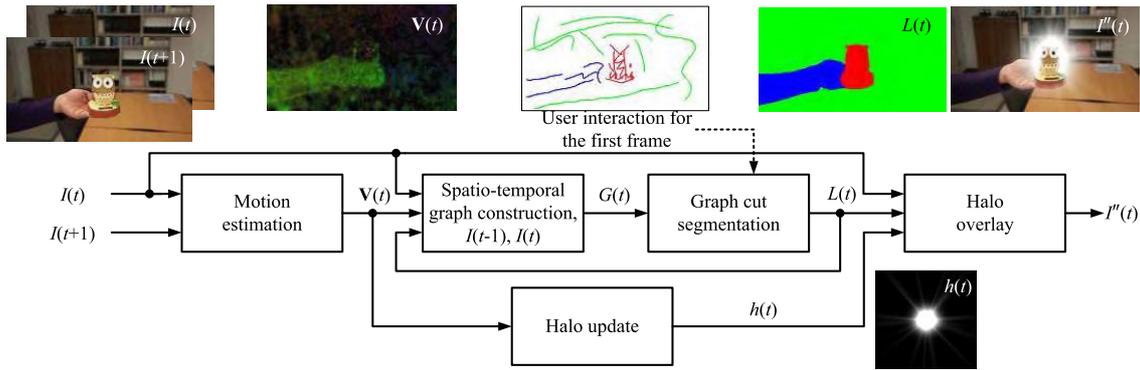


Fig. 1 Overview of the proposed halo-overlying algorithm.

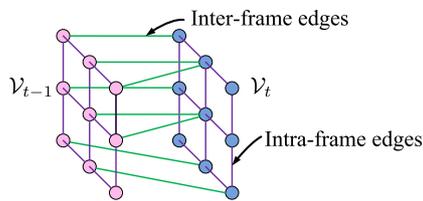


Fig. 2 Motion based spatio-temporal graph.

In addition, the notion of the local and the global motion is participated in data cost function for both  $\mathcal{V}_{t-1}$  and  $\mathcal{V}_t$ . While data cost for  $\mathcal{V}_{t-1}$  depends on  $\mathbf{V}(t-1)$ ,  $\mathbf{V}(t)$  is included in the data cost for  $\mathcal{V}_t$ . Global motion  $\mathbf{V}_{global}$  is average motion of nodes connected to nodes  $L_p^* = L_{BG}$  along the inter-frame edges.

To take the prescribed ideas into account, the proposed data cost function in Eq. (2) is formed by sum of two components: the location dependent cost  $N(\cdot)$  in Eq. (3) and motion dependent cost  $\gamma(\cdot)$  in Eq. (4).

$$E_{data}(L_p) = \begin{cases} w_{prev}\gamma^*(p) + w_{seed}N(p), & p \in \mathcal{V}_{t-1}, L_p = L_p^* \\ w_{prev}\gamma^*(p) + w_{seed}, & p \in \mathcal{V}_{t-1}, L_p \neq L_p^* \\ w_{curr}\gamma(p), & p \in \mathcal{V}_t \end{cases} \quad (2)$$

where  $w_{seed}$ ,  $w_{prev}$ , and  $w_{curr}$  are weights for seeds,  $\mathcal{V}_{t-1}$ , and  $\mathcal{V}_t$ , respectively.

$$N(p) = \exp\left[-\frac{d^2(p)}{2(\sigma_b d_{max}(L_p^*))^2}\right] \quad (3)$$

$$\gamma(p) = \begin{cases} 1 - \left(\frac{|\mathbf{V}_{global} - \mathbf{V}_p|}{\max|\mathbf{V}_{global} - \mathbf{V}_p|}\right)^\alpha, & L_p \in L_{FG} \\ \left(\frac{|\mathbf{V}_{global} - \mathbf{V}_p|}{\max|\mathbf{V}_{global} - \mathbf{V}_p|}\right)^\alpha, & L_p = L_{BG} \end{cases} \quad (4)$$

$N(\cdot)$  reflects low confidence of near boundary nodes and  $\sigma_b$  controls the size of the region. Distance from label boundary to a node  $p$  is measured by quasi-Euclidean function  $d(p)$  [5] and is normalized by the maximum distance

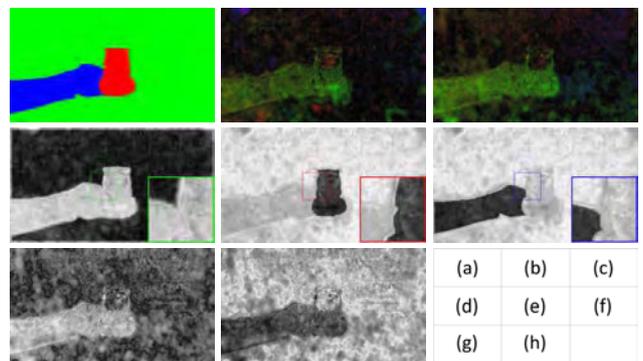


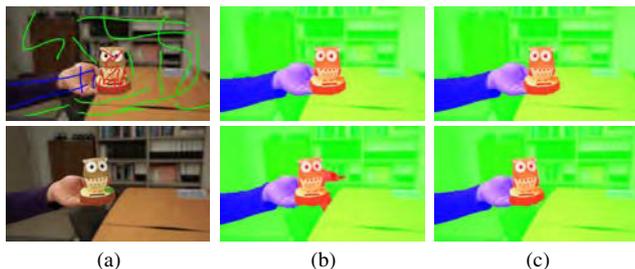
Fig. 3 (a)  $L^*$ , (b)  $\mathbf{V}(t-1)$ , (c)  $\mathbf{V}(t)$ ; data cost for  $L_{BG}$  (d) and  $L_{FG1}$  (e), and (f)  $L_{FG2}$  of  $\mathcal{V}_{t-1}$ ; data cost for  $L_{BG}$  (g) and (h)  $L_{FG1}$  &  $L_{FG2}$  of  $\mathcal{V}_t$ .

$d_{max} = \max_{L_p \in L} d(p)$ . Moreover, a node with distinct motion from  $\mathbf{V}_{global}$  is likely to be assigned  $L_{FG_i}$  through the motion dependent cost  $\gamma(p)$ . The vector  $\mathbf{V}_p$  in Eq. (4) denotes the motion vector of a node  $p$  and  $\alpha$  is a user defined exponent.

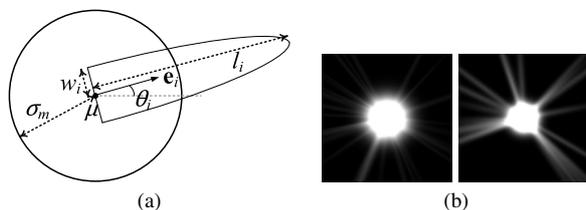
Figure 3 illustrates the data costs described in Eq. (2).  $L_{FG1}$ ,  $L_{FG2}$ , and  $L_{BG}$  are represented in red, blue, and green, respectively. Motion vectors in Fig. 3 (b) and (c) are displayed by color coding representation [6]. Increase of the data cost around boundary for node in  $\mathcal{V}_{t-1}$  is shown in Fig. 3 (d)~(f). Since the data cost of  $\mathcal{V}_t$  is calculated based on  $\mathbf{V}(t)$  only,  $L_{FG1}$  and  $L_{FG2}$  share the identical data cost as in Fig. 3 (h).

### 2.3 Smoothness Cost

Color coherency of nodes connected by any edge needs to be preserved. Motion coherency, on the other hand, is not guaranteed for an inter-frame edge, since an abrupt change in motion is quite commonplace between frames. Therefore, the proposed smoothness cost function is defined as Eq. (5), where  $w_{color}$  and  $w_{motion}$  are weights for the color and the motion, respectively. And  $\mathbf{C}_p$  represents the color vector of a node  $p$ . The indicator function  $\lambda(p, q)$  is given as Eq. (6).



**Fig. 4** (a) Selected frame of video sequence 'Owl', (b) Corresponding segmentation results without considering confidence of  $L^*$ , (c) Segmentation result by the proposed energy function.



**Fig. 5** (a) Proposed halo model by mixing Gaussian and hemi-Gaussians, and (b) examples of synthesized halos.

$$E_{smooth}(L_p, L_q) = \begin{cases} w_{color} \frac{1}{|C_p - C_q|} - w_{motion} \frac{\lambda(p, q)}{|\mathbf{v}_p - \mathbf{v}_q|}, & L_p \neq L_q \\ 0, & L_p = L_q \end{cases} \quad (5)$$

$$\lambda(p, q) = \begin{cases} 1, & p, q \in \mathcal{V}_t \text{ or } p, q \in \mathcal{V}_{t-1} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

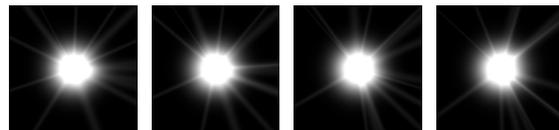
Final segmentation results of selected frames from a video sequence Owl are presented in Fig. 4. The input frames including user interaction for the first frame are in the left column. In the middle column, we can observe that fully trusting  $L^*$  leads to false segmentation and its propagation over frames. However, in the right column, despite of several motion components within and among layers, the proposed method achieves fair results by suppressing the propagation of errors.

### 3. Virtual Halo Effect

#### 3.1 Halo Synthesis

The halo effect is synthesized as Eq. (7), mixture of an isotropic Gaussian and  $N_r$  hemi-Gaussians of Eq. (8). Main circular body of a halo is centered at  $\boldsymbol{\mu}$  and its size is determined by user-defined  $\sigma_m$  in Eq. (7). Brightness gain factor  $G$  needs to be very large for realistic saturation of pixel intensity.

$N_r$  emitting subordinate rays are elongated in axial directions as in Fig. 5. A hemi-Gaussian, as an emitting subordinate ray, elongates in direction of a unit vector  $\boldsymbol{\theta}_i \in [0, 2\pi)$ ,  $\theta_i \in [0, 2\pi)$  with length  $l_i$  and width  $w_i$ . Roles of the parameters and examples of synthesized halos are depicted in Fig. 5.



**Fig. 6** Evolution of halo over time by modifying the parameters.

$$h(\mathbf{x}) = G \left[ \frac{1}{2} \exp\left(-\frac{|\mathbf{x} - \boldsymbol{\mu}|^2}{\sigma_m^2}\right) + \frac{1}{2N} \sum_{i=1}^{N_r} g(\mathbf{x} - \boldsymbol{\mu}, \mathbf{e}_i, l_i, w_i) \right] \quad (7)$$

$$g(\mathbf{x}, \mathbf{e}_i, l_i, w_i) = \begin{cases} \exp\left[-\left(\left|\frac{\mathbf{x} \cdot \mathbf{e}_i}{l_i}\right|^2 + \left|\frac{\mathbf{x} \times \mathbf{e}_i}{w_i}\right|^2\right)\right], & \mathbf{x} \cdot \mathbf{e}_i \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

Temporal transformation of the halo is simulated by updating the parameters as  $\phi_i \leftarrow \phi_i + \delta\phi_i$ , where  $\phi_i \in \{l_i, w_i, \theta_i\}$ . The parameters  $l_i, w_i, \theta_i$  and their temporal variations  $\delta l_i, \delta w_i, \delta \theta_i$  are initialized randomly within limited ranges. When the magnitude of the parameter exceeds the range, it and its temporal variation are reinitialized to stay in the range.

Location of the halo  $\boldsymbol{\mu}$  for the first frame is defined by the user and updated using the estimated motion. Since motion estimation errors are caused around object boundaries by occlusion and disocclusion, motion of the halo is computed with a bilateral filter assigning more weight to the center of the halo and inner region of the object. Update of  $\boldsymbol{\mu}$  and the bilateral weight  $\eta(\mathbf{x})$  are given in Eqs. (9) and (10), respectively.

$$\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} + \sum_{\mathbf{x} \in \text{obj}} \eta(\mathbf{x}) \mathbf{V}(\mathbf{x}) / \sum_{\mathbf{x} \in \text{obj}} \eta(\mathbf{x}) \quad (9)$$

$$\eta(\mathbf{x}) = \exp\left(-\frac{|\mathbf{x} - \boldsymbol{\mu}|^2}{\sigma_1^2}\right) \left[ 1 - \exp\left(-\frac{d^2(\mathbf{x})}{\sigma_2^2}\right) \right] \quad (10)$$

As shown in Fig. 6, temporal variations of the halos are perceptually convincing to be suitable for video applications.

#### 3.2 Overlaying the Halo

A halo is overlaid through two steps. Firstly,  $I(\mathbf{x})$  is blended with  $h(\mathbf{x})$  straightforwardly to generate  $I'(\mathbf{x})$  as in Fig. 7 (a) by Eq. (11).  $I'(\mathbf{x})$ , however, is not natural and shows artifacts around the object boundary.

$$I'(\mathbf{x}) = \begin{cases} (1 - h(\mathbf{x}))I(\mathbf{x}) + h(\mathbf{x})I_{\max}, & L(\mathbf{x}) \neq L_{FG1}, h(\mathbf{x}) < 1 \\ I_{\max}, & L(\mathbf{x}) \neq L_{FG1}, h(\mathbf{x}) \geq 1 \\ I(\mathbf{x}), & L(\mathbf{x}) = L_{FG1} \end{cases} \quad (11)$$

In Eq. (11),  $I_{\max}$  denotes the saturated pixel intensity, i.e.,  $I_{\max} = 255$  for 8-bit encoded images.

Secondly, blooming phenomenon [7] is applied to create a natural and visually pleasing image. A new halo-overlaid image  $I''(\mathbf{x})$  in Fig. 7 (b) is obtained by diffusing



**Fig. 7** Halo-overlaid images (a)  $I'$ : without halo diffusion process (b)  $I''$ : with the halo diffusion.

**Table 1** Computational time for  $640 \times 360$  video sequences.

Function blocks	Time (sec./frame)
motion estimation	1.33
graph-cut segmentation	2.49
synthesis and overlay of the halo	1.17
Total	4.99



**Fig. 8** Selected frames of halo-overlaid video sequences.

$h(\mathbf{x})$  in the background region toward the foreground as Eq. (12).

$$I''(\mathbf{x}) = \begin{cases} (1 - h'(\mathbf{x}))I(\mathbf{x}) + h'(\mathbf{x})I_{\max}, & L(\mathbf{x}) = L_{FG1}, h'(\mathbf{x}) < 1 \\ I_{\max}, & L(\mathbf{x}) = L_{FG1}, h'(\mathbf{x}) \geq 1 \\ I'(\mathbf{x}), & L(\mathbf{x}) \neq L_{FG1} \end{cases} \quad (12)$$

$$h'(\mathbf{x}) = \max_{L(\mathbf{x}')=L_{BG}} \exp\left(-\frac{|\mathbf{x}' - \mathbf{x}|}{2\sigma_d^2}\right) h(\mathbf{x}) \quad (13)$$

The diffused halo map  $h'(\mathbf{x})$  is defined as Eq. (13) and the diffusion strength is controlled by a parameter  $\sigma_d$ .

#### 4. Experimental Results

Figure 8 demonstrates halo-overlaid real-world sequences by the proposed method. Despite of cluttered background and complex motions of camera and objects, the synthesized halos are merged into video frames without visible artifacts. The third column of Fig. 8 shows that colored halos can be synthesized by simply assigning different gain factor  $G$  to

each color channel.

Processing time of the proposed algorithm is shown in Table 1. The time was measured in a PC with Intel Core i7 3.2 GHz and 6 GB RAM. The motion is estimated using multigrid scheme [8] for speed-up. Almost half of time is spent in graph-cut segmentation since the labeling process is performed for two consecutive frames to avoid error propagation. The synthesis and overlaying the halo consumes also large amount of time due to several per-pixel computations of exponentials.

#### 5. Conclusion and Future Works

In this letter, we present a scheme of overlaying halo effect as a new visual experience by highlighting an object of interest in a video sequence. The requirements of user interaction is limited to the first frame. The halo is overlaid automatically for the following frames by utilizing graph-cut based video segmentation and motion estimation algorithm. Our static and dynamic modeling of halos produces realistic and visually pleasing effects in video sequences. Although the work described in this letter is devoted to a video editing, the proposed algorithm is instantly applicable to a still image editing by excluding the motion estimation.

In a point of user-convenience, introduction of object detectors can be very helpful for implementation of fully-automated halo effects without user intervention.

An additional improvement in quality of overlaid halo is expected by applying detailed modeling of an optical phenomenon, such as a lens flare.

#### References

- [1] 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, pp.1124–1137, 2004.
- [2] K.N. Ngan, Video segmentation and its applications, Springer, New York, 2011.
- [3] A. Delong, A. Osokin, H. Isack, and Y. Boykov, "Fast approximate energy minimization with label costs," Int. J. Comput. Vis., vol.96, no.1, pp.1–27, 2012.
- [4] A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnörr, "A multi-grid platform for real-time motion computation with discontinuity-preserving variational methods," Int. J. Comput. Vis., vol.70, no.3, pp.257–277, 2006.
- [5] M. Sonka, V. Hlavac, and R. Boyle, Image processing, analysis, and machine vision, 3rd ed., Thompson Learning, Toronto, 2008.
- [6] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi, "Bilateral filtering-based optical flow estimation with occlusion detection," Proc. European Conference on Computer Vision, pp.211–224, 2006.
- [7] G.C. Holst and T.S. Lomheim, CMOS/CCD sensors and camera systems, JCD Publishing; SPIE, Portland, 2007.
- [8] U. Trottenberg, C.W. Oosterlee, and A. Schüller, Multigrid, Academic Press, San Diego, 2001.