Image Segmentation Using Iterated Graph Cuts Based on Multi-scale Smoothing

Tomoyuki Nagahashi[1], Hironobu Fujiyoshi[1], and Takeo Kanade[2]

[1] Dept. of Computer Science, Chubu University.
Matsumoto 1200, Kasugai, Aichi, 487-8501 Japan.
kida@vision.cs.chubu.ac.jp, hf@cs.chubu.ac.jp
http://www.vision.cs.chubu.ac.jp

[2] The Robotics Institute, Carnegie Mellon University.
Pittsburgh, Pennsylvania, 15213-3890 USA.
tk@cs.cmu.edu

Abstract. We present a novel approach to image segmentation using iterated Graph Cuts based on multi-scale smoothing. We compute the prior probability obtained by the likelihood from a color histogram and a distance transform using the segmentation results from graph cuts in the previous process, and set the probability as the t-link of the graph for the next process. The proposed method can segment the regions of an object with a stepwise process from global to local segmentation by iterating the graph-cuts process with Gaussian smoothing using different values for the standard deviation. We demonstrate that we can obtain 4.7% better segmentation than that with the conventional approach.

1 Introduction

Image segmentation is a technique of removing objects in an image from their background. The segmentation result is typically composed on a different background to create a new scene. Since the breakthrough of Geman and Geman [1], probabilistic inference has been a powerful tool for image processing.

The graph-cuts technique proposed by Boykov [2][3] has been used in recent years for interactive segmentation in 2D and 3D. Rother *et al.* proposed GrabCut[4], which is an iterative approach to image segmentation based on graph cuts. The inclusion of color information in the graph-cut algorithm and an iterative-learning approach increases its robustness. However, it is difficult to segment images that have a complex edge. This is because it is difficult to achieve segmentation by overlapping local edges that influence the cost of the n-link, which is calculated from neighboring pixels. Therefore, we introduced a coarse-to-fine approach to detecting boundaries using graph cuts.

We present a novel method of image segmentation using iterated Graph Cuts based on multi-scale smoothing in this paper. We computed the prior probability obtained by the likelihood from a color histogram and a distance transform, and set the probability as the t-link of the graph for the next process using the segmentation results from the graph cuts in the previous process. The proposed method could segment regions of an object with a stepwise process from global to local segmentation by iterating the graph-cuts process with Gaussian smoothing using different values for the standard deviation.

2 Graph Cuts

This section describes the graph-cuts-based segmentation proposed by Boykov and Jolly[2].

2.1 Graph Cuts for Image Segmentation

An image segmentation problem can be posed as a binary labeling problem. Assume that the image is a graph, $G = (V, E)$, where V is the set of all nodes and E is the set of all arcs connecting adjacent nodes. The nodes are usually pixels, p, on the image, P, and the arcs have adjacency relationships with four or eight connections between neighboring pixels, $q \in N$. The labeling problem is to assign a unique label, L_i, to each node, $i \in V$, i.e. $L_i \in \{\text{"}obj\text{"}, \text{"}bkg\text{"}\}$. The solution, $\boldsymbol{L} = \{L_1, L_2, \ldots, L_p, \ldots, L_{|P|}\}$, can be obtained by minimizing the Gibbs energy, $E(\boldsymbol{L})$:

$$E(\boldsymbol{L}) = \lambda \cdot R(\boldsymbol{L}) + B(\boldsymbol{L}) \tag{1}$$

where

$$R(\boldsymbol{L}) = \sum_{p \in P} R_p(L_p) \tag{2}$$

$$B(\boldsymbol{L}) = \sum_{\{p,q\} \in N} B_{\{p,q\}} \cdot \delta(L_p, L_q) \tag{3}$$

and

$$\delta(L_p, L_q) = \begin{cases} 1 \text{ if } & L_p \neq L_q \\ 0 \text{ otherwise.} & \end{cases} \tag{4}$$

The coefficient, $\lambda \geq 0$, in (1) specifies the relative importance of the region-properties term, $R(\boldsymbol{L})$, versus the boundary-properties term, $B(\boldsymbol{L})$. The regional term, $R(\boldsymbol{L})$, assumes that the individual penalties for assigning pixel p to "*obj*" and "*bkg*", corresponding to $R_p(\text{"}obj\text{"})$ and $R_p(\text{"}bkg\text{"})$, are given. For example, $R_p(\cdot)$ may reflect on how the intensity of pixel p fits into a known intensity model (e.g., histogram) of the object and background. The term, $B(\boldsymbol{L})$, comprises the "boundary" properties of segmentation $\boldsymbol{L}$. Coefficient $B_{\{p,q\}} \geq 0$ should be interpreted as a penalty for discontinuity between p and q. $B_{\{p,q\}}$ is normally large when pixels p and q are similar (e.g., in intensity) and $B_{\{p,q\}}$ is close to zero when these two differ greatly. The penalty, $B_{\{p,q\}}$, can also decrease as a function of distance between p and q. Costs $B_{\{p,q\}}$ may be based on the local intensity gradient, Laplacian zero-crossing, gradient direction, and other criteria.

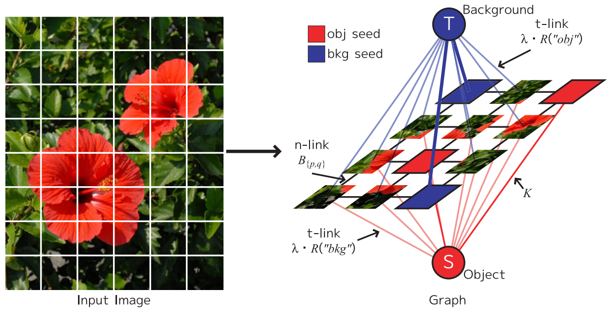

Fig. 1. Example of graph from image

Table 1. Edge cost

Edge		Cost	For
n-link	$\{p, q\}$	$B_{\{p,q\}}$	$\{p, q\} \in N$
t-link	$\{p, S\}$	$\lambda \cdot R_p(\text{"}bkg\text{"})$	$p \in P, p \notin \mathcal{O} \cup \mathcal{B}$
		K	$p \in \mathcal{O}$
		0	$p \in \mathcal{B}$
	$\{p, T\}$	$\lambda \cdot R_p(\text{"}obj\text{"})$	$p \in P, p \notin \mathcal{O} \cup \mathcal{B}$
		0	$p \in \mathcal{O}$
		K	$p \in \mathcal{B}$

Figure 1 shows an example of a graph from an input image. Table 1 lists the weights of edges of the graph. The region term and boundary term in Table 1 are calculated by

$$R_p(\text{“}obj\text{”}) = -\ln \Pr(I_p|\mathcal{O}) \tag{5}$$

$$R_p(\text{“}bkg\text{”}) = -\ln \Pr(I_p|\mathcal{B}) \tag{6}$$

$$B_{\{p,q\}} \propto \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p, q)} \tag{7}$$

$$K = 1 + \max_{p \in P} \sum_{q:\{p,q\} \in N} B_{\{p,q\}}. \tag{8}$$

Let $\mathcal{O}$ and $\mathcal{B}$ define the “object” and “background” seeds. Seeds are given by the user. The boundary between the object and the background is segmented by finding the minimum cost cut [5] on the graph, G.

2.2 Problems with Graph Cuts

It is difficult to segment images including complex edges in interactive graph cuts [2], [3], as shown in Fig. 2. This is because the cost of the n-link is larger than that of the t-link. If a t-link value is larger than that of an n-link, the number of error pixels will be increased due to the influence of the color. The edge has a strong influence when there is a large n-link. The cost of the n-link between the flower and the leaf is larger than that between the leaf and the shadow as seen in Fig. 2. Therefore, it is difficult to segment an image that has a complex edge.

We therefore introduced a coarse-to-fine approach to detect boundaries using graph cuts.

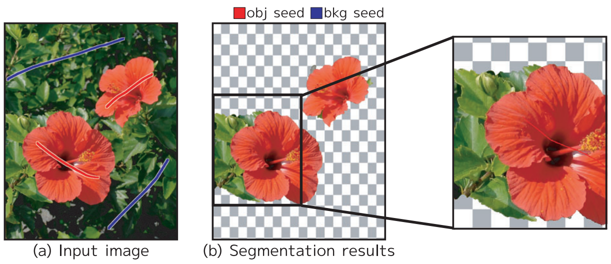

Fig. 2. Example of poor results

3 Iterated Graph Cuts by Multi-scale Smoothing

We present a novel approach to segmenting images using iterated Graph Cuts based on multi-scale smoothing. We computed the prior probability obtained by the likelihood from a color histogram and a distance transform, and set the probability as the t-link of the graph for the next process using the segmentation results from graph cuts in the previous process. The proposed method could segment regions of an object using a stepwise process from global to local segmentation by iterating the graph-cuts process with Gaussian smoothing using different values the for standard deviation.

3.1 Overview of proposed method

Our approach is outlined in Fig. 3. First, the seeds the "foreground" and "back-

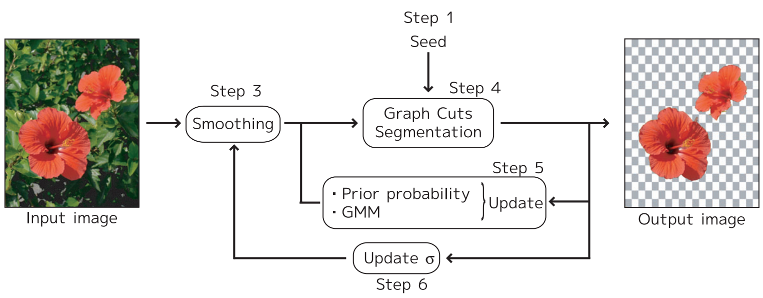

Fig. 3. Overview of proposed method

ground" are given by the user. The first smoothing parameter, σ, is then determined. Graph cuts are done to segment the image into an object or a background. The Gaussian Mixture Model (GMM) is then used to make a color distribution model for object and background classes from the segmentation results obtained by graph cuts. The prior probability is updated from the distance transform by the object and background classes of GMM. The t-links for the next graph-cuts process are calculated as a posterior probability which is computed a prior probability and GMMs, and σ is updated as, $\sigma = \alpha \cdot \sigma (0 < \alpha < 1)$. These processes are repeated until $\sigma = 1$ or classification converges if $\sigma < 1$.

The processes are as follows.

Step 1 Input seeds
Step 2 Initialize σ
Step 3 Smooth the input image by Gaussian filtering
Step 4 Do graph cuts
Step 5 Calculate the posterior probability from the segmentation results and set as the t-link
Step 6 Steps 1-5 are repeated until $\sigma = 1$ or classification converges if $\sigma < 1$.

The proposed method can be used to segment regions of the object with a stepwise process from global to local segmentation by iterating the graph-cuts process with Gaussian smoothing using different values for the standard deviation, as shown in Fig. 4.

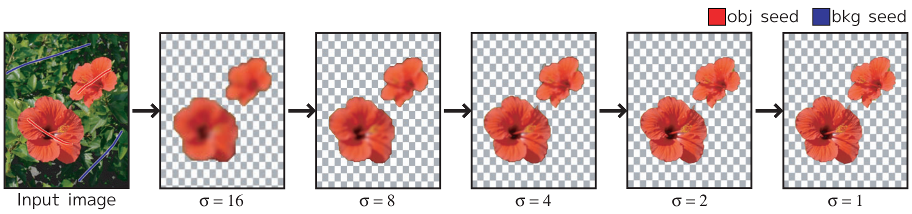

Fig. 4. Example of iterating the graph-cuts process

3.2 Smoothing image using down sampling

The smoothing image is created with a Gaussian filter. Let I denote an image and $G(\sigma)$ denote a Gaussian kernel. Smoothing image $L(\sigma)$ is given by

$$L(\sigma) = G(\sigma) * I. \tag{9}$$

If Gaussian parameter σ is large, it is necessary to enlarge the window size for the filter. As it is very difficult to design such a large window for Gaussian filtering. We used down-sampling to obtain a smoothing image that maintained the continuity of σ.

Smoothing image $L_1(\sigma)$ is first computed using the input image I_1 increasing σ. Image I_2 is then down-sampled into half the size of input image I. Smoothing image $L_2(\sigma)$ is computed using the I_2. Here, the relationship between $L_1(\sigma)$ and $L_2(\sigma)$

$$L_1(2\sigma) \fallingdotseq L_2(\sigma) \tag{10}$$

We obtain the smoothing image, which maintains continuity of σ without changing the window size, using this relationship. Figure 5 shows the smoothing process obtained by down-sampling. The smoothing procedure was repeated until $\sigma = 1$ in our implementation.

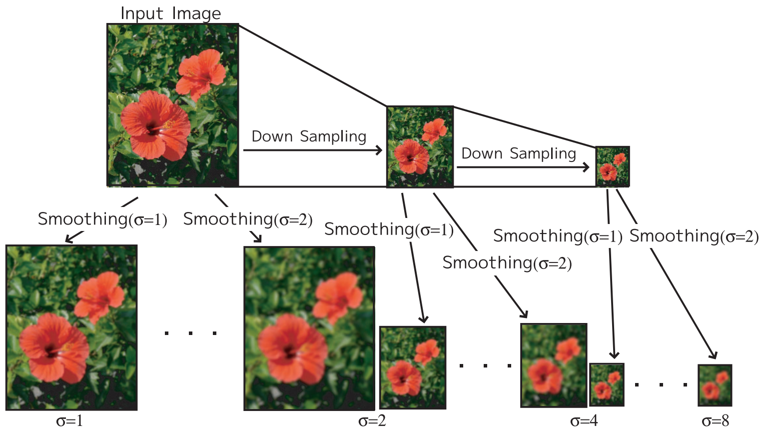

Fig. 5. Smoothing Image using down-sampling

3.3 Iterated Graph Cuts

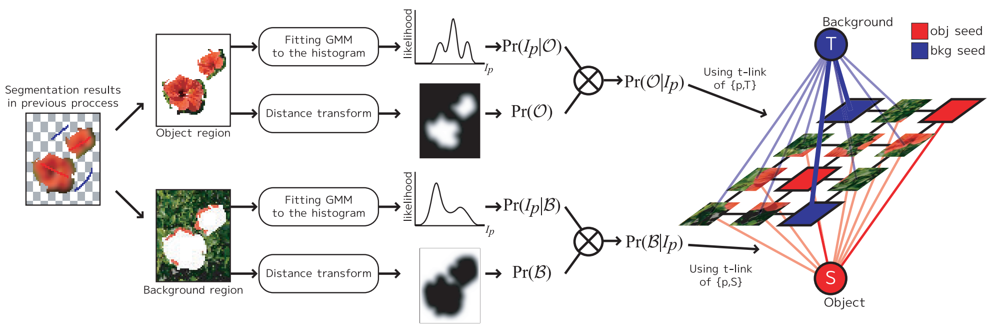

Fig. 6. Outline of updating for likelihood and prior probability

We compute the prior probability obtained by the likelihood from a color histogram and a distance transform using the segmentation results from the graph cuts in the previous process, and set the probability as the t-link using

$$R'_p(\text{"}obj\text{"}) = -\ln \Pr(\mathcal{O}|I_p) \tag{11}$$

$$R'_p(\text{"}bkg\text{"}) = -\ln \Pr(\mathcal{B}|I_p) \tag{12}$$

where $\Pr(\mathcal{O}|I_p)$ and $\Pr(\mathcal{B}|I_p)$ is given by

$$\Pr(\mathcal{O}|I_p) = \frac{\Pr(\mathcal{O})\Pr(I_p|\mathcal{O})}{\Pr(I_p)} \tag{13}$$

$$\Pr(\mathcal{B}|I_p) = \frac{\Pr(\mathcal{B})\Pr(I_p|\mathcal{B})}{\Pr(I_p)}. \tag{14}$$

$\Pr(I_p|\mathcal{O}), \Pr(I_p|\mathcal{B})$ and $\Pr(\mathcal{O}), \Pr(\mathcal{B})$ are computed from the segmentation results using graph cuts in the previous process. Figure 6 outlines t-link updating obtained by the likelihood and prior probability.

Updating likelihood The likelihoods $\Pr(I_p|\mathcal{O})$ and $\Pr(I_p|\mathcal{B})$ are computed by GMM[6]. The GMM for RGB color space is obtained by

$$\Pr(I_p|\cdot) = \sum_{i=1}^{K} \alpha_i p_i(I_p|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{15}$$

$$p(I_p|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{3/2}|\Sigma|^{1/2}} \cdot \exp\left(\frac{1}{2}(I_p - \boldsymbol{\mu})^T \Sigma^{-1}(I_p - \boldsymbol{\mu})\right). \tag{16}$$

We used the EM algorithm to fit the GMM[7].

Updating prior probability The prior probabilities $\Pr(\mathcal{O})$ and $\Pr(\mathcal{B})$ are updated by spatial information from graph cuts in the previous process. The next segmentation label is uncertain in the vicinity of the boundary. Therefore, the prior probability is updated by using the results of a distance transform. The distance from the boundary is normalized from 0.5 to 1. Let d_{obj} denote the distance transform of the object, and d_{bkg} denote the distance transform of the background. The prior probability is given by:

$$\Pr(\mathcal{O}) = \begin{cases} d_{obj} & \text{if } d_{obj} \geq d_{bkg} \\ 1 - d_{bkg} & \text{if } d_{obj} < d_{bkg} \end{cases} \tag{17}$$

$$\Pr(\mathcal{B}) = 1 - \Pr(\mathcal{O}) \tag{18}$$

Finally, using $\Pr(I_p|\mathcal{O}), \Pr(I_p|\mathcal{B})$ from GMM, and $\Pr(\mathcal{O})$ and $\Pr(\mathcal{B})$ from distance transform, posterior probability is computed by means of Eq. (11) and (12). We compute a prior probability obtained by the likelihood from a color histogram and a distance transform, and set the probability as the t-link of the graph for the next process using the segmentation results obtained by graph cuts in the previous process.

Figure 7 shows examples of segmentation results when the n-link is changed. When σ is small, the boundary-properties term, $B_{\{p,q\}}$, at the object is small because of the complex texture. Therefore, graph-cuts results do not work well for image segmentation. However, $B_{\{p,q\}}$ in the smoothing image is small between the object and background. The proposed method can used to segment regions of the object using a stepwise process from global to local segmentation by iterating the graph-cuts process with Gaussian smoothing using different values for the standard deviation.

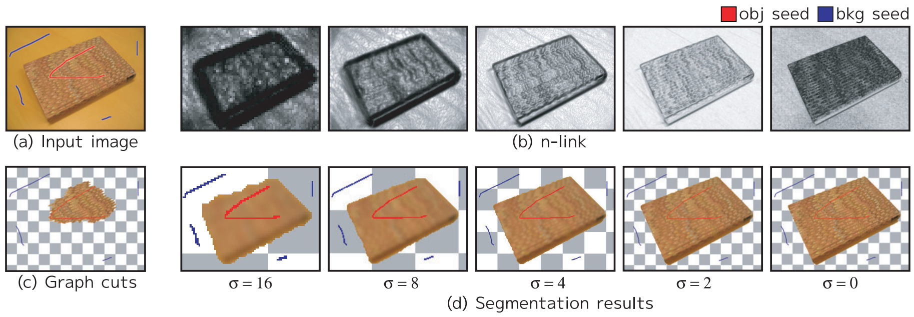

Fig. 7. Example of segmentation results when changing n-link

4 Experimental results

4.1 Dataset

We used 50 images(humans, animals, and landscapes) provided by the GrabCut database [8]. We compared the proposed method, Interactive Graph Cuts[2] and GrabCut[4] using the same seeds. The segmentation error rate is defined as

$$\text{over segmentation} = \frac{\text{object of miss detection pixels}}{\text{image size}} \tag{19}$$

$$\text{under segmentation} = \frac{\text{background of miss detection pixels}}{\text{image size}}. \tag{20}$$

4.2 Experimental Results

Table 2 lists the error rate (%) for segmentation results using the proposed method and the conventional methods [2], [4]. The proposed method can obtain

Table 2. Error rate[%]

	Interactive Graph Cuts[2]	GrabCut[4]	Proposed method
Over segmentation	1.86	3.33	1.12
Under segmentation	1.89	1.59	0.49
total	3.75	4.93	1.61

2.14% better segmentation than Interactive Graph Cuts. To clarify the differences between the methods, successfully segmented images were defined, based on the results of interactive Graph Cuts, as those with error rates below 2%, and missed images were defined as those with error rates over 2%. Table 3 list the segmentation results for successfully segmented and missed images. The proposed method and Interactive graph cuts are comparable in the number of successfully segmented images. However, we can see that the proposed method can obtain 4.79% better segmentation than Interactive Graph Cuts in missed images. The proposed method can be used to segment regions of the object using a stepwise process from global to local segmentation. Figure 8 shows examples of segmentation results obtained with the new method.

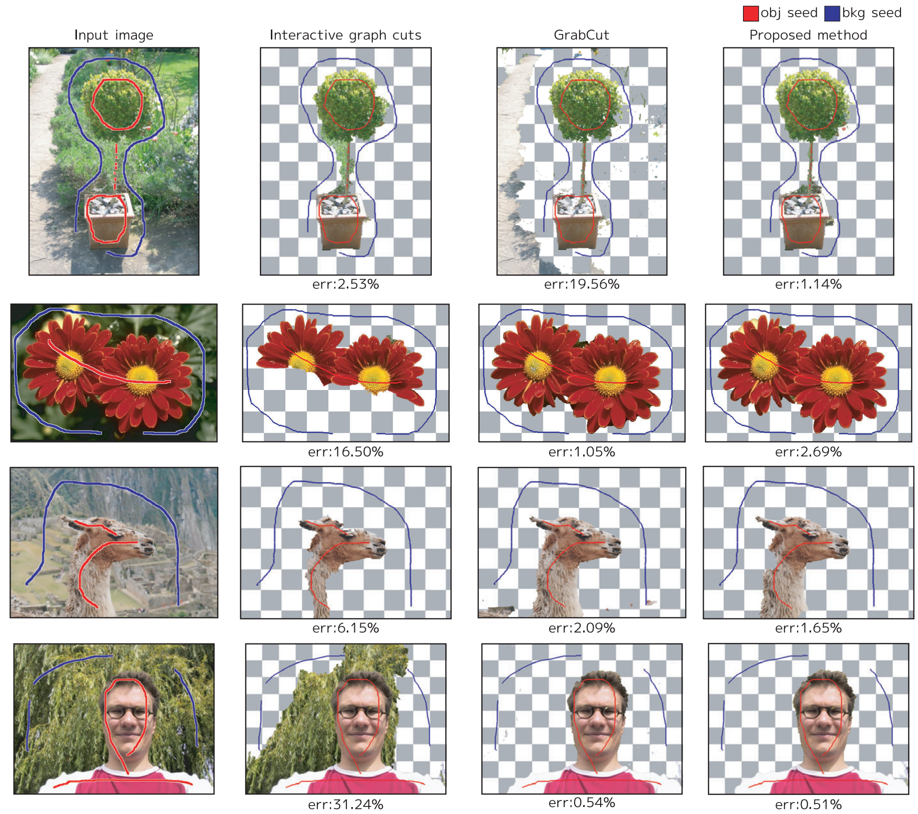

Fig. 8. Examples of segmentation results

4.3 Video segmentation

The proposed method can be applied to segmenting N-D data. A sequence of 40 frames (320x240) was treated as a single 3D volume. A seed is given to the first frame. Figure 9 shows examples of video segmentation obtained with the new method. It is clear that the method we propose can easily be applied to segmenting videos. We can obtain video-segmentation results.

5 Conclusion

We presented a novel approach to image segmentation using iterated Graph Cuts based on multi-scale smoothing. We computed the prior probability obtain by the likelihood from a color histogram and a distance transform, and set the probability as the t-link of the graph for the next process using the segmentation results from the graph cuts in the previous process. The proposed method could segment regions of an object with a stepwise process from global to local segmentation by iterating the graph cuts process with Gaussian smoothing using different values for the standard deviation. We demonstrated that we could obtain 4.7% better segmentation than that with the conventional approach.

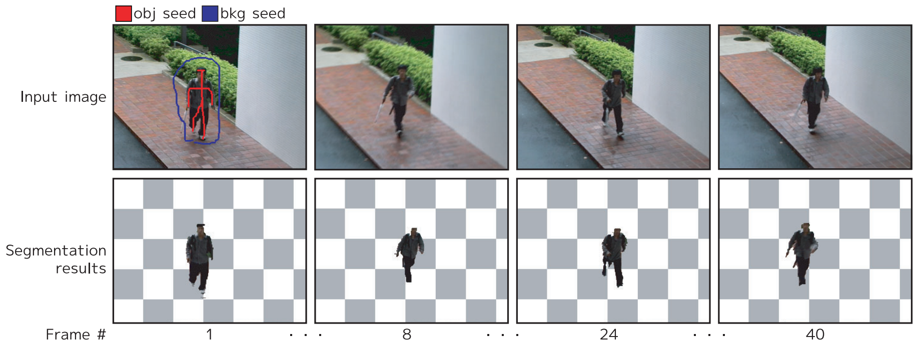

Fig. 9. Example of video segmentation.

Table 3. Error rate [%]

		Interactive Graph Cuts[2]	GrabCut[4]	Proposed method
Successfully segmented (26 images)	Over segmented	0.29	3.54	0.81
	Under segmented	0.43	1.03	0.22
	total	0.72	4.58	1.03
Missed images (24images)	Over segmented	3.56	3.10	1.45
	Under segmented	3.47	2.21	0.79
	total	7.04	5.31	2.25

Future works includes increased speed using super pixels and highly accurate video segmentation.

References

1. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. " *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 721-741, Nov. 1984.
2. Y. Boykov, and M-P. Jolly, "Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images" , *ICCV*, vol. I, pp. 105-112, 2001.
3. Y. Boykov, and G. Funka-Lea, "Graph Cuts and Efficient N-D Image Segmentation", *IJCV*, vol. 70, no. 2, pp. 109-131, 2006.
4. C. Rother, V. Kolmogorv, and A. Blake, ""GrabCut":Interactive Foreground Extraction Using Iterated Graph Cuts, " *ACM Trans. Graphics* (SIGGRAPH '04), vol.23, no.3, pp.309-314, 2004.
5. Y. Boykov, and V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision", *PAMI*, vol. 26, no. 9, pp. 1124-1137, Sept. 2004.
6. C. Stauffer and W.E.L Grimson, "Adaptive Background Mixture Models for Real-time Tracking", *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.246-252, 1999.
7. A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum-likelihood From Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society.* Series B, Vol. 39, No. 1 , pp. 1-38, 1977.
8. GrabCut Database: http://research.microsoft.com/vision/cambridge/i3l/segmentation/GrabCut.htm