AUTOMATIC SEGMENTATION OF SALIENT OBJECTS USING ITERATIVE REVERSIBLE GRAPH CUT

Chanho Jung, Beomjoon Kim and Changick Kim

Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST)
Daejeon, Republic of Korea
e-mail: peterjung, bjkim81@kaist.ac.kr and cikim@ee.kaist.ac.kr

ABSTRACT

There have been several interactive approaches to extracting objects from still images, since it is significantly difficult to automatically segment objects in complex background. In this paper, we present a novel automatic scheme for extracting salient objects from natural images. To this end, segmentation of salient objects is formulated as a global energy minimization problem in an iterative self-adaptive framework. By employing a saliency detection technique, object and background seeds are inferred automatically. The problem in this step is that the automatically generated seeds may not be reliably positioned. An iterative reversible graph cut method is introduced to overcome the problem inherent in the saliency-based seed extraction method. In the iterative self-adaptive framework, bidirectional state transitions are iteratively involved to reduce the misclassified pixels. Experimental results show that the proposed segmentation method yields more accurate segmentation results than previous segmentation approaches.

Keywords— Automatic Object Segmentation, Graph Cuts, Bidirectional State Transition, Iterative Refinement, Saliency-based Seed Extraction

1. INTRODUCTION

Object segmentation is one of the most important and challenging issues in image analysis and computer vision. It facilitates a number of object-driven high-level applications, such as object recognition and scene understanding. Interaction-aware methodologies have been the most widely used techniques for object segmentation. Mortensen et al. [1] proposed an interactive segmentation method using global graph search. The graph searching allows a user to choose minimum cost contour in an image. Bayes matting [2] requires a user-specified trimap, which separates an image into foreground, background, and unknown regions. Based on color distribution models, alpha values for the unknown region are evaluated. Graph cuts [3] are powerful optimization tools for the interactive object segmentation. In the graph cut algorithms, hard constraints are imposed by a user to provide seed positions and the goal is to find a minimum cost cut among all segmentations satisfying the given hard constraints. The graph cut algorithms yield robust segmentation results, even when the object and background can not be well separated. Rother et al. [4] extended the graph cut approaches to simplify user interaction. In the method, a user specifies a rectangle loosely around an object. However, there are some drawbacks on those interactive segmentation approaches [5]. Segmentation performance is extremely dependent on user-specified seed extraction. Thus, additional interactions are necessary when the seeds are not provided accurately. Moreover, the user interactions are time-consuming and often infeasible.

In order to develop fully automated application systems, user interaction should be excluded in the object segmentation step. First of all, the seeds need to be extracted automatically. The object segmentation results are dominated by the reliability of seed extraction in the automatic object segmentation, as well as in the interactive approach. There have been several techniques to detect salient regions in an image based on human visual system [6]-[8]. Typically, the salient regions are highly correlated with interesting objects in an image. Therefore, saliency detection can be helpful in extracting the seeds automatically [5].
However, it is significantly difficult to obtain reliable seeds by the saliency-based seed extraction. Figure 1 shows the inherent problems of automatic seed extraction. In Fig. 1, the salient regions are detected by a spectral residual approach [6] and the seeds are automatically extracted by a saliency cuts [5]. As shown in Fig. 1, the ill-positioned seed pixels are highly likely to yield unwanted segmentation results, since part of the object seed belongs to the background and vice versa, and thus the classification error in seed region is propagated to non-seed region. Thus, robust segmentation results can not be achieved. Although, in [5], object-oriented morphological operation is employed to refine the extracted seeds, it is still difficult to get robust segmentation results. Figure 2 shows the segmentation results obtained by [5].

In this paper, to overcome the problem above-mentioned, we propose an iterative reversible graph cut method. The segmentation of salient objects is treated as a minimal cut problem in an iterative self-adaptive framework. In order to reduce the misclassified pixels, bidirectional state transitions between the object and background are iteratively considered based on learned probabilistic color distribution models. Our main contributions are as follows:

- **Self-adaptive Bidirectional State Transition**: The object pixels misclassified to the background and vice versa in the seed and non-seed regions can be correctly classified in the self-adaptive manner.
- **Iterative Refinement**: In contrast to “GrabCut” [4] in which iterative optimization has been applied to allow incomplete labeling, the proposed iterative refinement is developed to address the inherent problems from incorrect labeling. In other words, the iterative scheme is involved to simplify substantial amount of user interaction in [4], whereas our segmentation is iteratively refined to get an optimal solution minimizing the classification error in the self-adaptive framework.

Moreover, the problems of interactive segmentation approaches can also be solved by our iterative self-adaptive framework. In other words, the proposed iterative reversible graph cut provides robust segmentation results even if the user-specified seeds are mislocated.

The rest of the paper is organized as follows: In section 2, the foundation on which the iterative reversible graph cut is developed is described. In section 3, the proposed method for automatically extracting salient objects is presented. Experimental results and conclusion are given in section 4.

2. IMAGE SEGMENTATION BY GRAPH CUTS

In this section, we briefly review the graph cuts based image segmentation [3] and discuss the problems we may have when it is applied to automated systems. For the image segmentation, an image is represented as an undirected graph $G = < V, E >$. The graph consists of a set of nodes $V$ and a set of undirected edges $E$ connect neighboring nodes. The set of nodes $V$ includes object terminal $S$ and background terminal $T$. The set of undirected edges $E$ is divided by two edge types: $n$-links and $t$-links. A cut is a subset of edges $C \subseteq E$ by which the terminals are separated on the generated graph $G(C) = < V, E \setminus C >$. The goal is to find an optimal cut minimizing the sum of edge costs as follows.

$$|C| = \sum_{e \in C} w_e,$$  \hspace{1cm} (1)

where $e = \{p, q\}$ and $w_e$ represent the edge and cost of $e$, respectively. It should be noted that the $n$-links and $t$-links on $C$ reflect boundary and regional properties of the segmentation, respectively.

2.1. Formulation

Let $P$ and $N$ denote a set of pixels in an image and a set of all unordered pairs $\{p, q\}$ of neighboring pixels in $P$. Let $A = (A_1, ..., A_p, ..., A_{|P|})$ denote a binary vector whose components designate assignments to $O$ or $B$, where $O$ and $B$ represent object and background, respectively. In order to estimate the globally optimal cut, a segmentation cost function is defined in terms of the boundary and regional properties of segmentation as follows.

$$E(A) = \lambda \cdot R(A) + B(A),$$  \hspace{1cm} (2)

where $R$ and $B$ represent the regional and boundary terms, respectively. In (2), $\lambda$ represents a coefficient which controls relative importance between the regional and boundary terms. The regional and boundary terms are defined as follows.

$$R(A) = \sum_{p \in P} R_p(A_p),$$  \hspace{1cm} (3)

$$B(A) = \sum_{\{p, q\} \in N} B_{\{p, q\}} \cdot \delta(A_p, A_q),$$  \hspace{1cm} (4)

where $R_p$ represents penalties for assigning pixel $p$ to $O$ or $B$, and $B_{\{p, q\}}$ represents a penalty for discontinuity between $p$ and $q$. In (4), $\delta(A_p, A_q)$ is zero if and only if $A_p$ and $A_q$ are equal.

2.2. Hard Constraints Driven Segmentation

One of the most important features of the graph cuts is initialization based on incorporating hard constraints as follows.

$$\forall p \in O, \hspace{1cm} A_p = O, \hspace{1cm} (5)$$

$$\forall p \in B, \hspace{1cm} A_p = B,$$  \hspace{1cm} (6)
from the mislocated seeds become more serious when the seeds are located automatically (see Fig. 1(c)). In [9], an efficient method for misclassification correction is introduced. However, the correction can not be allowed in the fully automatic strategy since it also requires user interaction. In the next section, we explain how to address the inherent problems within the proposed iterative reversible graph cut framework.

3. PROPOSED METHOD: TOWARD AN AUTOMATIC SEGMENTATION

In order to extract salient objects from a given natural image, two processes are consecutively involved: 1) the automatic seed extraction based on the saliency detection technique and 2) the graph cuts based image segmentation in the iterative self-adaptive framework.

3.1. Saliency-based Seed Extraction

Let \( I \) denote a given image. From the given image, spectral residual [6] is evaluated by

\[
\mathcal{R}(f) = \mathcal{L}(f) - \mathcal{A}(f),
\]

where \( \mathcal{L}(f) \) and \( \mathcal{A}(f) \) represent the log spectrum of \( I \) and the averaged spectrum, respectively. The saliency map is obtained by

\[
\mathcal{S}(x) = g(x) \ast \tilde{f}^{-1} [\exp(\mathcal{R}(f) + \Psi(f))]^2,
\]

where \( g(x) \) represents a Gaussian filter. \( \tilde{f}^{-1} \) and \( \Psi(f) \) represent the inverse Fourier transform and the phase spectrum of \( I \), respectively. In Fig. 1(b), the saliency maps obtained by (8) are illustrated. Given the saliency map, we can obtain the object map \( \mathcal{D}(x) \) by performing binarization [5], [6].

By employing the object map and a priori knowledge with respect to the regular location of salient objects [5], object seed map \( \mathcal{S}_O(x) \) and background seed map \( \mathcal{S}_B(x) \) are defined as follows.

\[
\mathcal{S}_O(x) = \begin{cases} 1 & \text{if } \mathcal{D}(x) = 1 \text{ and } r(x) \leq \rho W, \\ 0 & \text{otherwise} \end{cases}
\]

\[
\mathcal{S}_B(x) = \begin{cases} 1 & \text{if } \mathcal{D}(x) = 0 \text{ and } r(x) > \rho W, \\ 0 & \text{otherwise} \end{cases}
\]

where \( r(x) \) and \( W \) represent the horizontal distance to image center and the width of \( I \), respectively. \( \rho \) represents a proportion coefficient. In Fig. 1(c), the object and background seed maps obtained by (9) and (10) are illustrated. In our work, \( \rho \) is set to 0.38. Note that the segmentation performance is highly sensitive to \( \rho \) in [5], whereas the consideration of influence by \( \rho \) is not necessary in the proposed scheme, which will be explained in subsection 3.2.

3.2. Iterative Reversible Graph Cut

In order to overcome the problems from the saliency-based seed extraction proposed in [5], a given image is segmented...
Fig. 5. Self-adaptive bidirectional state transition scheme embedded in graph cut framework.

in the iterative self-adaptive framework. We first initialize the trimap which consists of $O$, $B$, and non-seed region. Note that $O$ and $B$ are obtained from $S_O$ and $S_B$, respectively. Then, the object and background color distributions are modeled as Gaussian mixtures with $K$ components based on the initialized trimap. Note that incorrect seed initialization leads to incorrect segmentation since the seeds are employed without correcting their topological errors in the classical graph cuts based approaches [3], [4], [9]. In this paper, the self-adaptive bidirectional state transition is embedded in the graph cut framework. Let $P(I_p|O)$ and $P(I_p|B)$ denote the likelihoods of pixel $p$ belonging to the given object and background Gaussian mixture models (GMMs), respectively. A two-state Markov chain model, which is illustrated in Fig. 5, is adopted based on the probabilistic color distributions (i.e., $P(I_p|O)$ and $P(I_p|B)$). In our work, the states indicate the object and background terminals (i.e., $S$ and $T$), since the transition is taken into account within the graph cut framework. Finally, the transition probability matrix is defined as follows.

$$P = \frac{1}{P(I_p|O) + P(I_p|B)} \begin{bmatrix} P(I_p|O) & P(I_p|B) \\ P(I_p|O) & P(I_p|B) \end{bmatrix}. \quad (11)$$

The bidirectional state transition gives a chance to adjust the misclassified pixels self-adaptively.

The proposed self-adaptive bidirectional state transition scheme is reflected on graph construction for combinatorial optimization. $V$ and $E$ are defined as follows [3], [9].

$$V = \mathcal{P} \cup \{S, T\}. \quad (12)$$

$$E = \mathcal{N} \bigcup_{p \in \mathcal{P}} \{\{p, S\}, \{p, T\}\}. \quad (13)$$

As we can see in (13), each pixel $p$ has two $t$-links $\{p, S\}$ and $\{p, T\}$. The graph cuts generate an optimal segmentation in terms of properties that are built into edge weights. Table 1 illustrates the weights of edges for embedding the self-adaptive bidirectional state transition. It should be noted that our edge weight assignment method, which is organized to allow the adjustment for all pixels $\mathcal{P}$, is opposed to those of the classical graph cuts based approaches. $R_p(O)$ and $R_p(B)$ are defined as follows by using negative log-likelihoods [3], [9].

$$R_p(O) = - \ln P(I_p|O). \quad (14)$$

$$R_p(B) = - \ln P(I_p|B). \quad (15)$$

### Table 1. Weights of edges in $E$ for embedding self-adaptive bidirectional state transition.

<table>
<thead>
<tr>
<th>edge</th>
<th>weight (cost)</th>
<th>for</th>
</tr>
</thead>
<tbody>
<tr>
<td>${p, q}$</td>
<td>$B_{(p,q)}$</td>
<td>${p, q} \in \mathcal{N}$</td>
</tr>
<tr>
<td>${p, S}$</td>
<td>$\lambda \cdot R_p(O)$</td>
<td>$p \in \mathcal{P}, p \notin O \cup B$</td>
</tr>
<tr>
<td></td>
<td>$\lambda \cdot R_p(O)$</td>
<td>$p \in O$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$p \in B$</td>
</tr>
<tr>
<td>${p, T}$</td>
<td>$\lambda \cdot R_p(B)$</td>
<td>$p \in \mathcal{P}, p \notin O \cup B$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$p \in O$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$p \in B$</td>
</tr>
</tbody>
</table>

Fig. 6. A simple segmentation example for a $4 \times 4$ image: (a) input image, (b) image with seeds, (c) segmentation result by classical graph cuts [3], [9], and (d) segmentation result when the proposed self-adaptive bidirectional state transition is applied. Note that incorrect labeling is found in (b).

By using max-flow algorithm [10], the minimum cost cut $\hat{C}$ is obtained. The self-adaptive bidirectional state transition between the object and background is performed as follows.

$$A_p = \begin{cases} O & \text{if } p \in B \text{ and } \{p, T\} \notin \hat{C} \\ B & \text{if } p \in O \text{ and } \{p, S\} \notin \hat{C} \end{cases}. \quad (16)$$

Note that this is well contrasted with the classical graph cuts based approaches. Figure 6 shows a simple segmentation example when the proposed self-adaptive bidirectional state transition is adopted. Note that the given image are incorrectly labeled as shown in Fig. 6(b). When the classical graph cut approach is employed, as shown in Fig. 6(c), the misclassification is observed due to incorrectly located seeds. However, the proposed scheme provides accurate segmentation result in spite of the incorrect labeling as shown in Fig. 6(d).

It is straightforward that the classification error in the seed region can be reduced by embedding the self-adaptive bidirectional transition into the graph cut framework as stated in (16). However, there are still the misclassified pixels in the non-seed region. To overcome the problems, the self-adaptive adjustment in (16) is extended to iterative refinement procedure in our work. Note that our iterative refinement reduces not only the misclassified pixels in the non-seed region but also those in the seed region, which might have not been adjusted by (16). Once the segmentation result is obtained by (16), it is iteratively refined by using (17), where $C_i$ denotes the minimum cost cut in the $i^{th}$ iteration. Note that $C_1$ corresponds to $\hat{C}$ in (16) and the segmentation of non-seed region (i.e., $p \notin O \cup B$) in the first iteration is determined by $C_1$. The proposed iterative self-adaptive bidirec-
Fig. 7. (a), (b) segmentation results of Fig. 1(a) by the proposed iterative reversible graph cut and (c) convergence of the proposed scheme.

A transitional state transition is performed for all pixels $P$ as follows.

$$A_p(C_i) = \begin{cases} 
\mathbb{O} & \text{if } A_p(C_{i-1}) = \mathbb{B} \text{ and } \{p, T\} \in C_i \\
\mathbb{B} & \text{if } A_p(C_{i-1}) = \mathbb{O} \text{ and } \{p, S\} \in C_i \\
A_p(C_{i-1}) & \text{otherwise} 
\end{cases}$$

for $i > 1$. (17)

Note that this is opposed to [4], since there is no constraint on the trimap in our work. While the segmentation is iteratively refined by (17), the object and background GMMs should also be updated to reflect the refined regional properties. Figure 7 shows the segmentation results obtained by the proposed iterative reversible graph cut. As shown in Fig. 7(a), the proposed segmentation method outperforms the conventional approach compared to Fig. 2. Fig. 7(b) shows the convergence property of our method. As shown in Fig. 7(b), the optimal cut can be obtained within a few iterations, even though some of automatically extracted seeds are not correctly located. The proposed overall iterative reversible graph cut procedure can be summarized as follows:

**Proposed Iterative Reversible Graph Cut Procedure:**

**Step 1:** initialize trimap

**Step 2:** build foreground and background GMMs from $O$ and $B$ respectively

**Step 3:** assign GMM components to $I_p$

**Step 4:** learn GMM parameters from $I$

**Step 5:** construct graph $g_i$ on which the self-adaptive bidirectional state transition scheme is reflected

**Step 6:** find minimum cost cut $C_i$ on $g_i$

**Step 7:** perform self-adaptive bidirectional state transition by using (16) or (17)

**Step 8:** repeat from **Step 3** to **7** until the number of state transitions is zero

Fig. 8. (a) input images, (b) segmentation results by [5], and (c) segmentation results by the proposed method.
In this paper, we have proposed a fully automated object segmentation technique. An iterative reversible graph cut is introduced to overcome the inherent problems from automatic seed extraction using saliency detection. Experimental results presented in this section prove that the proposed scheme outperforms the conventional methods.

5. REFERENCES


