

An Interactive Segmentation Method Based on Superpixel

Shu Yang*

School of Information Engineering, Communication University of China, Beijing, China

Yaping Zhu

*School of Information Engineering, Communication University of China, Beijing, China
Department of Computer Science, University of Copenhagen, Denmark*

Xiaoyu Wu

School of Information Engineering, Communication University of China, Beijing, China

ABSTRACT: This paper proposes an interactive image-segmentation method which is based on superpixel. To achieve fast segmentation, the method is used to establish a Graphcut model using superpixels as nodes, and a new energy function is proposed. Experimental results demonstrate that the authors' method has excellent performance in terms of segmentation accuracy and computation efficiency compared with other segmentation algorithm based on pixels.

Keywords: image segmentation; superpixel; Graphcut

1 INTRODUCTION

The idea of image segmentation is to distinguish pixels belonging to foreground or background. An interactive segmentation-method based on graph model was proposed by Boykov, Y.Y.[1] in 2001. Many related works have been done to improve segmentation accuracy and efficiency based on it. The innovation can be divided into two categories: energy function based methods [2] [3] and superpixel-based methods [4]. Energy function-based methods create new background/foreground models to compute energy function, such as Grabcut [2], it improves accuracy at the expense of efficiency. The superpixel-based methods, such as lazy snapping [4], achieve more excellent results than energy function-based methods. Besides, superpixel-based methods can be regarded as a preprocessing of co-segmentation [5] to process groups of images.

In this paper, we propose a superpixel-based method to improve both accuracy and efficiency of Graphcut[1]. We pre-segment the original image into small areas, and apply them to set up graph models for Graphcut. The main contributions of this paper include:

- 1) We improve algorithm Graphcut [2] with a superpixel generation method of which parameters can be selected through experiments.
- 2) A new energy function is created, and it is more suitable to superpixel-based model.
- 3) We compare our method with two state-of-the-art interaction segmentation methods on the CMU-Cornell iCoseg Dataset [5], as our method can be a preprocessing of co-segmentation task.

The rest of this paper is organized as follows. First,

*Corresponding author: yangshu91@163.com

some related algorithms according to image segmentation and superpixels production are quoted in the next section. Section 3 describes the whole framework from two steps: the pre-segmentation and the interactive segmentation. Experimental results and parameters' selection are illustrated in section 4 and the conclusion is given in the last section.

2 RELATED WORK

Image segmentation has been a popular research topic in the area of image processing. It can be divided into two categories: the automatic segmentation and the interactive segmentation.

The automatic methods are based on images' features, such as color, texture and edge, and the methods can be divided into the edge-based and the area-based. Edge-based methods detect edges with gray value of pixels, as step changing through edges which can be detected with differential operators, such as Canny, Sobel and Laplace operators. The representative of area-based method is Watershed, and it over-segments images into small area on condition that the edges of objects are retained. The disadvantage of Watershed is the disorder of area size, while Achanta [6] solved this problem. Achanta introduced a new area-based method "Simple Linear Iterative Clustering (SLIC)", using a clustering approach based on k-means to generate superpixels efficiently. Besides, SLIC uses CIELAB color space to replace gray histogram and the number of superpixels can be controlled by users.

Interactive segmentation algorithms build models of foreground and background with seed pixels given by users. Graphcut [1] is a typical interactive segmenta-

tion method. It builds a model named “Graphcut” with pixels being nodes. An energy function is defined using the distance between seed pixels and other pixels to calculate region properties and the distance among neighbor pixels to calculate boundary properties. The minimum cut method is adopted for partition nodes into two sets and achieves the final segmentation. Grabcut[2] is an improvement of Graphcut which adopts GMM in RGB color space instead of gray histogram. However, Grabcut improves segmentation accuracy at the expense of efficiency as it has iterative algorithm.

The lazy snapping [4] is combined with automatic and interactive segmentation methods to solve segmentation problem. It pre-segments images into superpixels with Watershed, and regards them as nodes to build the Graphcut model.

In this paper, the method of Lazy snapping [4] is adopted and improved with SLIC [6] replacing Watershed. The algorithm of SLIC produces superpixels with similar sizes and ordered location. Moreover, color space-based method is suitable to color feature extraction, and the feature will be used for our further research on co-segmentation [5]. Then, use these superpixels as nodes to set up the Graphcut model. In order to compute the energy function which is improved in CIELAB color space, we obtain the seed pixels of background and foreground given by users’ interaction and use the information of each superpixels’ neighborhood. Finally, the minimum cut algorithm is applied to the energy function to achieve the final segmentation.

3 PROBLEM FORMULATION

3.1 Image pre-segmentation

First, use the approach of Achanta [6] to over-segment the original image with fixed parameter which is found appropriate from experiments. We also present a straight forward function to extract the information of superpixels’ neighborhood which is of importance to Graphcut model.

3.1.1 Method of SLIC

Achanta [6] addresses the superpixel generation by “Simple linear iterative clustering (SLIC)” which is an adaptation of k-means. There are two parameters in this algorithm, K and m. K is the number of superpixels in one image, and m is the influence parameter of boundaries. K is also the number of clusters in k-means which is used to compute the regular grid interval S as Equation (1), while N is the number of pixels of original image. They are shown as follows:

$$S = \sqrt{N / K} \quad (1)$$

The idea of SLIC is iterative. First, initialize cluster centers $C_k=[l_k, a_k, b_k, x_k, y_k]$ in CIELAB color space by sampling pixels at regular grid steps S and move them to the lowest gradient position in a 3×3 neighborhood. Second, for each pixel in the $2S*2S$

region around C_k , compute the distance D between C_k and use Equation (2), and distribute pixels to different clusters by minimizing D. Third, find the new centers by computing mean $[l, a, b, x, y]$ in each cluster. Repeat the second and third steps until centers are relatively fixed.

$$D = \sqrt{\|C_k - i\|_{L_2}^2 + \left(\frac{d}{S}\right)^2 \times m^2} \quad (2)$$

$$d = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}$$

$$\|C_k - i\|_{L_2} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2}$$

3.1.2 Data of superpixels

Through experiment (see section 4.1), we look for appropriate parameters of SLIC and present a straight forward function to extract neighborhood’s information.

After the pre-segmentation, the information of superpixels is prepared for next step. For each superpixel, the useful information including the area size which is also the number of pixels belong to it; the cluster center $C_k=[l_k, a_k, b_k, x_k, y_k]$; the information of its neighborhood; and its flag Fk denotes whether it belongs to foreground or background. The area size and cluster center is obtained with SLIC. We propose a simple search method to obtain the information of neighborhood which is important to the energy function.

First, we label all the superpixels using L_k in range of $[0, K-1]$, and the pixels belonging to the superpixel k are given the same label. Then, the labels of pixels adjoining superpixel are regarded as the neighborhood information. Particularly, the repetitive labels are not supposed to be included. We transform superpixel’s problem to pixel’s problem, and obtain the neighborhood information of all superpixels by searching around all pixels as Algorithm 1. (x_i, y_i) is the location of pixel i and L_i is its center’s label (for example, $L_i=k$ denotes that pixel i belongs to superpixel k). We define four neighborhoods with location which are (x_i-1, y_i) , (x_i, y_i-1) , (x_i+1, y_i) and (x_i, y_i+1) as its “Left, Up, Right and Down”. N_k is the neighborhood information of superpixel k which is a string of non-repetitive labels. Particularly, as for images’ boundary or corner pixels, we only consider its three or two neighborhoods.

Algorithm 1:

```

Foreach pixel i do
1.  $k=L_i$ 
2. foreach pixelj being “Left, Up, Right and Down” of i do
   if  $L_j$  is not equal to k &&  $L_j$  is not in  $N_k$  then
3. add  $L_j$  into  $N_k$ 
   end if
end for
end for
    
```

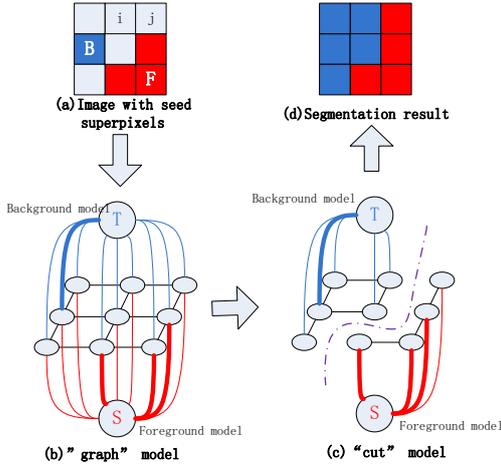


Figure 1. Graphcut model based on superpixel

3.2 Iterative segmentation

In this section, we propose an interactive method to obtain the final segmentation. We adopt Boykov and Jolly's [1] Graphcut model and use superpixels generated by SLIC as nodes shown in Figure 1(a) We pre-segment images into superpixels such as i and j , and extend the seed pixels given from user into seed superpixels. We use red to label foreground and blue to label background. (b) We build "graph" model using the superpixels as nodes, besides, there are another two nodes "T" and "S" respectively representing background model and foreground model. Every two nodes are linked by an edge with different weight, and the weight is defined by energy function. (c) A minimum cut classifies nodes into two classes. (d) The two groups of nodes denote the superpixels belonging to foreground or background.

To an image with size of 500×333 , there are about K nodes in our method instead of 500×333 nodes in Graphcut model[1]. As a consequence, the improved method also reduces the number of edges in the model and computing job. Moreover, a new energy function is proposed which is more suitable to superpixel-based model. The energy function consists of region property and boundary property as Equation (3). V is the set of all nodes, W is the set of adjacent nodes we obtained in last section, and α is the influence parameter of adjacent nodes. The region energy $E_1(F_i)$ is similar to the region energy of the lazy snapping [4] in Equation (4) and Equation (5), while the boundary energy is different as shown in Equation (6).

$$E = \sum_{i \in V} E_1(F_i) + \alpha \sum_{i, j \in W} E_2(F_i, F_j) \quad (3)$$

To compute E_1 , superpixels including seed pixels which are marked by users are regarded as seed superpixels. $E_1(F_i)$ is the cost when the flag of node i is F_i . For the non-seed superpixel i , d_i^f and d_i^b are the

minimum L2 distance from its center $C_i = [l_i, a_i, b_i]$ in each background and foreground superpixels' center. It is shown in Equation (5):

$$\begin{cases} d_k^f = \min \|C_k - i\|_{L_2} & \forall i \in \text{foreground seed} \\ d_k^b = \min \|C_k - i\|_{L_2} & \forall i \in \text{background seed} \end{cases} \quad (4)$$

$$\begin{cases} E_1(F_i=1) = 0 & E_1(F_i=0) = \infty & \forall i \in \text{foreground seed} \\ E_1(F_i=1) = \infty & E_1(F_i=0) = 0 & \forall i \in \text{background seed} \\ E_1(F_i=1) = \frac{d_i^f}{d_i^f + d_i^b} & E_1(F_i=0) = \frac{d_i^b}{d_i^f + d_i^b} & \forall i \in \text{not seed} \end{cases} \quad (5)$$

The boundary energy is represented as $E_2(F_i, F_j)$, the cost of the flag of adjacent nodes i and j are respectively F_i and F_j . $Dis(i, j)$ is the distance of two centers $C_i = [x_i, y_i]$ and $C_j = [x_j, y_j]$, while $\|C_i - C_j\|$ is the Euclidean distance of $C_i = [l_i, a_i, b_i]$ and $C_j = [l_j, a_j, b_j]$.

$$\begin{aligned} E_2(F_i, F_j) &= dis(i, j)^{-1} [F_i \neq F_j] \exp(-\beta \|C_i - C_j\|_{L_2}^2) \\ dis(i, j) &= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \\ \|C_i - C_j\|_{L_2} &= \sqrt{(l_i - l_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2} \end{aligned} \quad (6)$$

Finally, the minimum cut is operated to final segment, and we can also correct the result with further interaction.

4 EXPERIMENT

4.1 Parameter selection

Our method is tested with images in the CMU-Cornell iCoseg Dataset [5] which can support our future research on co-segmentation. In order to find appropriate parameters of SLIC, several images are segmented with our method. We compare the time of superpixels' generation and the accuracy of segmentation with different m and K . Particularly, we give fixed seed pixels on original images and use the intersection-over-union score [8] to measure the accuracy as Equation (7). M is the number of test images, G_j is the foreground pixels of our method, while GT_j is the true foreground pixels given by CMU-Cornell iCoseg Dataset [5]. $Area^{(G_j \cap GT_j)}$ is the number of pixels our result and the truth ground sharing.

$$Score = \frac{1}{M} \sum_{j=1}^M \frac{Area(G_j \cap GT_j)}{Area(G_j \cup GT_j)} \quad (7)$$

The boundaries' influence parameter m is in the range of [1, 80]. We test our method with fixed $K=400$

and varying m in Figure 2 and Figure 3. We show some results of segmentation in Figure 4 and find that when $m = 60$, though the score is great, the boundaries of objects are missed. As a result, we propose $m=20$.

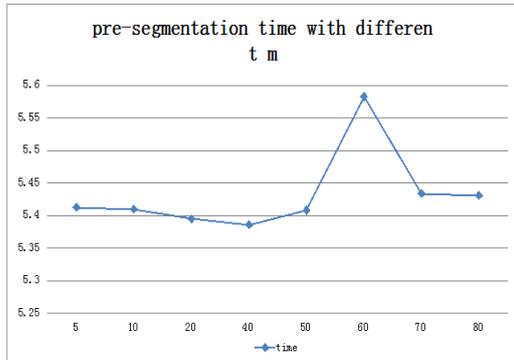


Figure 2. Pre-segmentation time with different m . The fixed parameter $K = 400$.

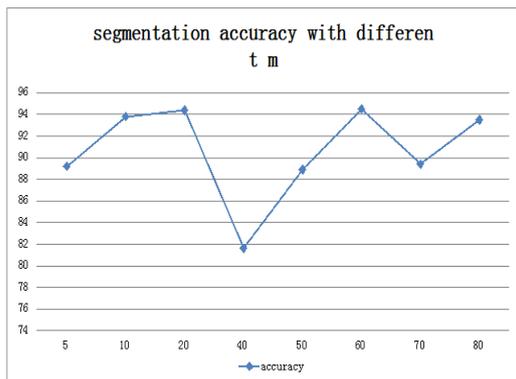


Figure 3. Segmentation accuracy with different m . The fixed parameter $K = 400$, and the seed pixels are same.

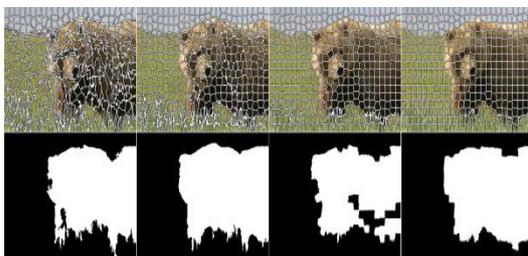


Figure 4. Segmentation result with $m=5, 20, 40$ and 60 . The fixed parameter $K = 400$, and the seed pixels are same.

Then, we test our method with different K , which is the number of superpixels in each image. The param-

eter m is fixed in 20. Figure 5 and Figure 6 show that the pre-segmentation time and segmentation accuracy are both increasing with the growth of K . In Figure 7, we find that when $K=500$, the segmentation result is good enough. Considering the larger superpixel including more features, which is better for our future research on co-segmentation, we select $K = 500$ instead of $K=1000$.

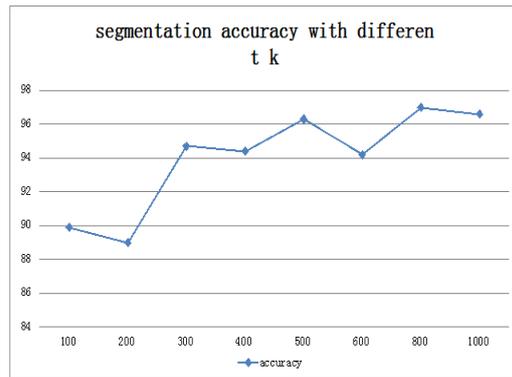


Figure 5. Pre-segmentation time with different K . The fixed parameter $m = 20$.

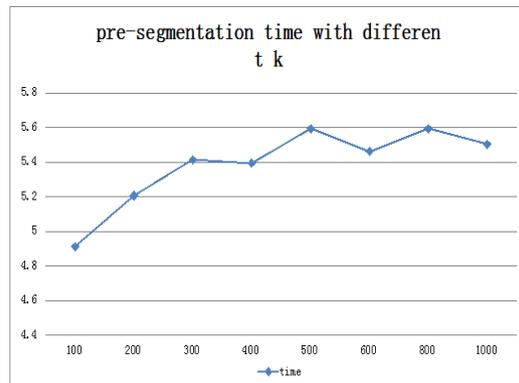


Figure 6. Segmentation accuracy with different K . The fixed parameter $m = 20$, and the seed pixels are same.

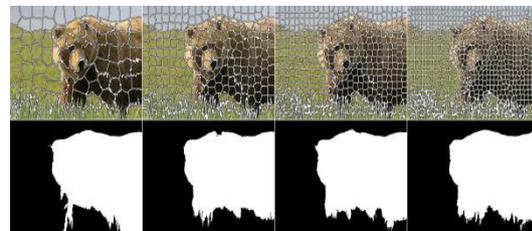


Figure 7. Segmentation result with different $K=100, 300, 500$ and 800 . The fixed parameter $m = 20$, and the seed pixels are same.

4.2 Segmentation result

Experiment shows that in some conditions, the proposed method performs better than other algorithms and various results are shown in Figure 8. We make comparison not only with Boykov's Graphcut [7], but also with another algorithm Grabcut [2], which is a state-of-the-art improvement of Graphcut and has been generally adopted.

With same seed pixels of background and foreground given by users, the new method does superior segmentation performance than original Graphcut, especially when the foreground is complex. It is obvious that some details are missed in Figure 8(c), and cannot segment out all objects. In this situation, more seed pixels are required from user.

The first interaction from user of Grabcut is a rectangle which includes the whole foreground showed in Figure 8(d) and it is simpler than seed pixels to a certain extent. However, when the area of foreground is huge, it will cause the size of rectangle which is similar to the whole image, thus the first interaction is useless. To obtain better result, seed pixels have to be used which consume more time and need more operations of user.

In contrast, our methods obtain better segmentation performance in the first interaction even when the foreground is very large or complex.

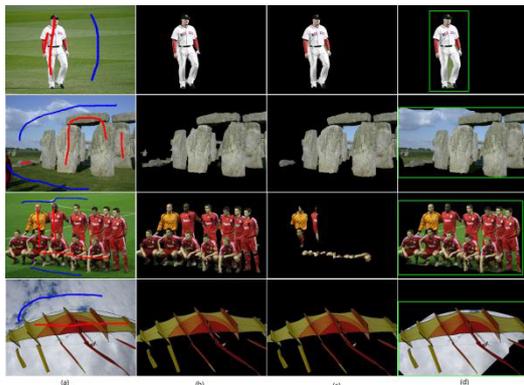


Figure 8. Comparison between our method, Graphcut and Grabcut. (a) original images with seed pixels of background (blue) and foreground (red) (b) Results of our method. (c) Results of Graphcut [7]. (d) Results of Grabcut [2].

4.3 Computation efficiency

The computation time is shown in Table 1. We compare our method with Graphcut [7] with same seed pixels and Grabcut [2] with initialization rectangle in the "First cut" column. Then we offer further seed pixels for all methods in the "Second cut" and "Third cut".

There are three factors: (1) The "First cut" column shows that the initialization efficiency of Graphcut [7] and Grabcut [2] is way less than our method. (2) For

Graphcut [7], the computation time is decreased with further cut, but it is still about 1 second, while our methods keep cut time around 0.22 second. (3) Most of the time, Grabcut's [2] first cut segmentation result is great. However, as mentioned earlier in section 4.2, when the area of foreground is huge, more cut will be needed and the computation time of further cut is similar to the initialization.

Table 1. Computation time.

Methods	Time(s)		
	First cut	Second cut	Third cut
Graphcut[7]	3.597	1.374	0.927
Grabcut[2]	5.218	3.045	3.179
Our method	0.224	0.222	0.230

5 CONCLUSIONS AND FUTURE WORK

The authors propose a method of improvement for Graphcut-based image segmentation. This method applies superpixel produced with SLIC algorithm to decrease nodes of Graphcut model. Meanwhile, an improved energy function is proposed which is more suitable to superpixel-based model. The authors' method is easier to implement and more effective when the foreground is large or complex.

Future work may be used to apply superpixel to extract discriminative features for co-segmentation [5]. When foreground of a group of related images are similar, we can extract the feature of superpixels from these images and apply machine learning methods to deal with co-segmentation task. Superpixel is more suitable for Bag-of-Words model to represent images. And the proposed method can perform preprocessing in co-segmentation.

ACKNOWLEDGEMENT

This paper is sponsored by National Natural Science Foundation of China (GN: 61101165).

REFERENCES

- [1] Boykov, Y.Y. 2001. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images, *International Conference on Computer Vision (ICCV)*, 1: 105 – 112.
- [2] Rother, C., Kolmogorov, V. & Blake, A. 2004. "GrabCut"-Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3).
- [3] Kolmogorov, V. & Zabini, R. 2004. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence (PAMI)*, *IEEE Transactions on*, 26(2).
- [4] Li, T., Sun, J., Tang, C.K. & Shum, I.Y. 2004. Lazy snapping, *ACM Transactions on Graphics*. 23(3).

- [5] Batra, D., Kowdle, A., Parikh, D., Jiebo Luo & Tsuhan Chen. 2010. iCoseg: Interactive co-segmentation with intelligent scribble guidance, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp: 3169 – 3176.
- [6] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P. & Süsstrunk, S. 2012. SLIC superpixels compared to state-of-the-art superpixel methods, *Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on*. 34(11): 888-905.
- [7] Meng T., Gorelick, L., Veksler, O. & Boykov, Y. 2013. GrabCut in one cut, *International Conference on Computer Vision (ICCV)*, pp: 1-8.
- [8] Everingham M. & Gool, L.V. 2010. The pascal visual object classes (VOC) challenge, *International Journal of Computer Vision*, 88(2): 303-338.