A Method for Image Co-segmentation with Seeded Foreground

Jiaji Wang*, Zhenjiang Miao, Tianyu Zhou, Qiang Zhang and Miaomiao Li

Institute of Information Science, School of Computer and Information Technology, Beijing Jiaotong University, Haidian District, 100044, Beijing, China

*E-mail: 12112069@bjtu.edu.cn

Abstract. Image segmentation is a key technology from image processing to analysis. Without proper segmentation, it is impossible to recognize correctly. In this paper, we propose a method for image co-segmentation based on the biased normalized cuts using a semi-supervised way to deal with foreground regions. In order to take advantage of biased normalized cuts to solve problem, we use 2D adaptive Wiener filter to smooth the seeded parts of images, then divide images into a set of super-pixels, after that take super-pixels as vertices to form a weighted undirected graph. Thus, the co-segmentation can be seen as an issue of graph partition that solved by biased normalized cuts. The experiments on image data sets show the superior performance of our method.

1. Introduction

Image segmentation is an important preprocessing for machine vision and image recognition. In general, people are only interested in foreground region of images. So, the image segmentation is very significant. It can divide the image into regions with different properties and extract the target corresponding to user's interest. In images, the parts that people interested are often called foregrounds.

Compared with single image segmentation, co-segmentation can segment a series of images for the similar targets simultaneously. The similar targets can be described as follows: in images, different objects belonging to the same type or an object observed from different directions. The co-segmentation can take advantage of the similarity between multiple foregrounds [1]. So, in contrast to the single ones, co-segmentation can get more information between the images and obtain more accurate results. Co-segmentation is widely used in image retrieval, object tracking, human motion capture, large-scale object recognition, and so on [2].

In our work, it is only need to mark foreground in one image, and then the method deals with a group of images automatically. The number of images is arbitrary, including only one image. In order to focus on a local region, a scrawl is used to mark the region of interest. 2D adaptive Wiener filter is served to smooth the seeded region and keep the region edge clear. After that, divide the images into super-pixel blocks and consider the blocks as vertices. So the images are changed into a graph with weighted and undirected edges. Now, the co-segmentation can be processed as an issue of graph partition and can be calculated by biased normalized cuts. Word “biased” here expresses that the method focuses on a part of images. The scrawl is used to mark the foreground, and it is the only interaction in our method. Biased normalized cuts can do a good job in this situation. The results show
that the method is efficient and robust. Sometimes, maybe several images contain no foregrounds of interest, and our approach is still effective.

This paper is organized as follows: Section 2 describes related works on unsupervised and supervised co-segmentation. Section 3 introduces our seeded foreground co-segmentation method. In Section 4, we show the experiment results of our method. Finally, a conclusion of this article is presented in Section 5.

2. Related work
In recent years, the methods of co-segmentation have attracted wide attention. Rother et al. [3] proposed the first co-segmentation method, which is used to segment a pair of images with similar foreground. It utilizes the traditional Markov Random Fields (MRF) model and the penalty term of foreground region. Mu and Zhou [4] extended Rother’s method by increasing a quadratic global constraint. Mukherjee et al. [5] proposed an algorithm based on the half-integrality. Hochbaum and Singh [6] improved the color histogram feature constraint, and used maximum flow algorithm based on graph theory. Vicente et al. [7] presented a dual decomposition algorithm, after that a co-segmentation algorithm based on the target object [8] was proposed. The method [8] divided the image into some local regions and extracted the common area from the regions. It used co-saliency map [9] as prior knowledge adding to the traditional MRF energy function. Then it optimized the method by graph-cuts. Methods [3-9] belong to unsupervised co-segmentation methods. This kind of methods have two drawbacks. The first one is that if the background of images are similar, in other words, the images are overall similarity, the methods will consider the background as foreground and can not work well. The second is that these methods are commonly used to deal with a pair of images, so they can not handle more images simultaneously.

Schnitman et al. [10] proposed a method that first segmented one image from a data set, and then generated classifiers to segment other images. Joulin et al. [11] used discriminative clustering method to achieve the combination of unsupervised discriminant and supervised classifier. Batra et al. [12] proposed an interactive co-segmentation method which need scribbles to label the images. Methods [10-12] belong to supervised co-segmentation methods. In contrast to the unsupervised ones, these ones are more complex, but can deal with more pictures and obtain better results.

3. Method of seeded foreground co-segmentation
Suppose there are $n$ images, and each contains $m$ pixels. The pixels are regarded as vertices and there are edges between vertices, then the images can be changed into a connected graph [13]. As a result, the co-segmentation can be considered as an issue of graph partition. However, with the number of images increases, the required storage space grows exponentially. This shortcoming limits the number of images in co-segmentation method. To break this constraint, we cluster the pixels into several groups. Each group is seen as a super-pixel [14], and then super-pixels are used as the vertices.

3.1. Image smoothing and generating super-pixels
In an image, one super-pixel is a small area which is composed of a series of adjacent pixels with similar position, color, brightness, texture and so on. In order to make a super-pixel belongs to either foreground or background, we take advantage of the seeded region to smooth the foreground. In this work, there is a manual interaction which draws a scribble in one image to mark the interesting region. After that, use simple linear iterative clustering method [14] to put the regions together, and then use a two dimensional adaptive Wiener filter to smooth the region and keep the edge clear. Using super-pixels instead of image pixels can reduce the number of vertices greatly. Besides, the method that deals with super-pixels is insensitive to noise.

Assuming that the size of each super-pixel is approximately the same and the size is roughly equal to $m_s$, so an image has about $n/m_s$ super-pixels and the distance between two neighboring super-pixels is about:
\[ d = \sqrt{m_s} \]  

(1)

3.2. Graph model

In a graph, there are vertices and edges. Super-pixels are vertices, and the relationship between two super-pixels can be quantized as a numerical value that treated as the weight of edge.

For two super-pixels, they may come from one image or two different images. For the first case, in one image, if two super-pixels are close to each other, they will be closely related. In other words, the corresponding edge has a large weight. In contrary, the edge will have a small weight. Between two super-pixels, when the distance \( D \) is larger than \( 3d \) (see formula 1), the relationship is considered small and the corresponding edge can be ignored. For the other case, two super-pixels come from different images, we set an edge between every two super-pixels because of requiring more information. If there are more edges, the corresponding matrices of the graph will be more accurate, too.

For two super-pixels \( s_a \) and \( s_b \), the histogram of HSV color space is used to measure the distance and MR8fast texture [15] is the feature vector. Define \( d(s_a, s_b) \) as:

\[
d(s_a, s_b) = \frac{1}{2} \sum_n \frac{(T_a(n) - T_b(n))^2}{T_a(n) + T_b(n)}
\]

(2)

where \( T(n) \) represents the \( n \)-th feature. When there exists an edge between \( s_a \) and \( s_b \), the weight \( w_{ab} \) is:

\[
w_{ab} = \exp\left(-d(s_a, s_b)^2/2\delta^2\right)
\]

(3)

where \( \delta \) is a constant. When there is no edge, the value of \( w_{ab} \) is set to 0.

3.3. Biased normalized cuts

For a graph \( G = (V, E) \), the edge weights can form a matrix \( M \). Assuming the amount of vertices \( |V| = c \), then the size of matrix \( M \) is \( c \times c \). Edge weight \( w_{ab} \) measures the similarity of vertex \( a \) and \( b \). For a set of vertices, the volume is defined as:

\[
\text{vol}(P) = \sum_{a \in P, b \notin V} w_{ab}
\]

(4)

Assuming \( P \subseteq V \), and \( Q \) is the complementary set of \( P \) (\( Q = V - P \)), then a cut between set \( P \) and \( Q \) is defined as:

\[
\text{cut}(P, Q) = \sum_{a \in P, b \in Q} w_{ab}
\]

(5)

Assuming \( H \) is the diagonal matrix of \( M \), and \( L \) is the normalized Laplacian matrix of \( M \), then elements of \( H \) are:

\[
e_{ab} = \sum_{a \in V} w_{ab}
\]

(6)

where \( a, b = 1, 2, ..., c \). In this situation, vector \( u \) satisfies \( u^\top H 1 = 0 \) and \( u^\top H u = 1 \), where \( u^\top \) is the transpose of \( u \). Then the biased normalized cuts is described as:

\[
\begin{aligned}
\min_x & \, x^\top L x \quad \text{s.t.} \quad u^\top H u = 1, u^\top H 1 = 0, (x^\top H u)^2 \geq k \\
\end{aligned}
\]

(7)

where \( k \) is a constant that keeps the angle between vector \( x \) and vector \( u \) in a certain range. In order to calculate the solution for formula 8, we combine the optimal result of normalized cuts [16] with vector \( u \). Then the optimal solution of our biased normalized cuts is:

\[
x^* = f \sum_{i=2}^K \frac{1}{\lambda_i - \gamma} h_i^\top H u = f \sum_{i=2}^K g_i h_i
\]

(8)
where value \( f \) is a constant, \( \lambda_i (i=2, \ldots, K) \) is the \( i \)-th smallest eigenvalue of matrix \( L \), \( K \) is the amount of eigenvalues, \( h_i \) represents the eigenvector corresponding to \( \lambda_i \), \( g_i \) represents the weight for each eigenvector, and \( \gamma \in (-\infty, \lambda_2) \).

4. Experiments and results

In order to check the performance, we do experiments on the iCoseg data set [12]. The data set contains 38 groups of images. For a group, each image contains one or more objects.

We first compare our approach with a basic method that using eigenvectors of the Laplacian matrix. The basic method is for the whole image, and does not sensitive to any local parts. An experiment is done on the “goose” image set. There are 10 images with a resolution of 500×333 pixels. The results are shown in figure 1 with NO. 1, 3, 6, 9 images. From figure 1(b) to (d), the results are respected to the second, third and fourth eigenvectors. Among these results, it is difficult to find a set of useful one. In our work, we roughly mark the area of goose in the first original image. The red scrawl is the input seed, which is used to let the approach focus on the parts of goose. Figure 1(f) presents the co-segmentation results by our approach.

Compared with the results of eigenvectors, ours are better especially in the body parts of goose. For the segmentation results of the second eigenvector, in the second line, there are no white gooses in the result. For the results of the third eigenvector, from the first to fourth line, there are only mouths in the results. For the fourth eigenvector, in the second and fourth line, the eyes are disappeared. These weak points do not appear in our results.

![Figure 1](image1.png)

**Figure 1.** Results of the proposed method and the second, third, fourth eigenvectors. (a) Original images, number 1, 3, 6, 9 images. (b)-(d) Results of second, third and fourth eigenvectors. (e) Super-pixels of images. (f) Results of our method.

![Figure 2](image2.png)

**Figure 2.** Co-segmentation results of Meng’s method and our method. The first and fourth lines are the original images. The second and fifth rows are Meng’s co-segmentation results. The third and sixth lines are the results of ours.
The second experiment is to compare our approach with Meng’s co-segmentation method [17]. Meng’s method uses saliency information and utilizes the shortest path to solve the problem. Experiment results of method [17] and our approach are shown in figure 2. The first three lines are about “skating” data set. The first line is original images, the second line is Meng’s results and the third line is ours (next three lines are also arranged in this way). For the outcomes in [17], the second and fifth results each contain a big piece of background, which leads to bad consequences. While, our results do not have this drawback. The last three lines are about “helicopter” data set. Results in [17] can not segment the propeller of helicopter, which is a shortcoming. In the fourth and fifth results, for the fuselage of helicopter, method in [17] are better than ours. Our shortcoming is that regarding several small pieces of background as the foreground.

In order to quantify the experiment results, we use correct rate and error rate to compare the two co-segmentation results with the ground truth. Two quantitative indicators are described as follows:

\[
\text{correct rate} = \frac{N(Q^f_x) + N(Q^g_y)}{N(\text{total})}
\]

\[
\text{error rate} = \frac{N(Q^f_x) + N(Q^g_y)}{N(\text{total})}
\]

where \(N(*)\) represents the amount of pixels, and “total” is all of the pixels in the image. In \(Q^f_x\), \(x\) belongs to either background \(B\) or foreground \(F\) which is judged by the co-segmentation results, and \(y\) is the ground truth. Formula 12 represents the correct rate that dividing the correct number of points by the total number, and formula 13 represents the error rate that dividing the wrong number of points by the total number. Quantitative indicators for “goose”, “skating” and “helicopter” data sets are shown in table 1. From the quantitative data, only in “goose” image set, the result of [17] is a little better than ours. However, for another two image sets and the average consequences, ours are better.

| Dataset   | Method  | Correct rate | Error rate |
|-----------|---------|--------------|------------|
|           |         | mean         | std        | mean       | std        |
| “goose”   | [17]    | 0.9894       | 0.0109     | 0.0106     | 0.0109     |
|           | Ours    | 0.9866       | 0.0080     | 0.0134     | 0.0080     |
|           | [17]    | 0.9838       | 0.0092     | 0.0162     | 0.0092     |
| “skating” | Ours    | 0.9903       | 0.0044     | 0.0097     | 0.0044     |
| “helicopter” | [17]  | 0.9815       | 0.0205     | 0.0185     | 0.0205     |
|           | Ours    | 0.9949       | 0.0022     | 0.0051     | 0.0022     |
| Average   | [17]    | 0.9849       | 0.0041     | 0.0151     | 0.0041     |
|           | Ours    | 0.9906       | 0.0042     | 0.0094     | 0.0042     |

5. Conclusion
In this paper, we presented an interactive seeded foreground method for image co-segmentation. The only simple interaction is to mark the interesting parts with a scrawl, in other words, mark out the seeded parts. In order to smooth the seeded region and keep the edge of the region clearly, we use the two dimensional adaptive Wiener filter to smooth the seeded parts in all the images. The smooth process can be conducive to get a good result of super-pixels. And super-pixels change the co-segmentation into a graph partition issue. In the graph, the seeded parts help the method focus on the interesting region. At last, the biased normalized cuts calculate the result of graph partition and segment the foreground region. Experimental consequences on several image sets show the superiority of the proposed method.
Acknowledgments
This work is supported by the NSFC 61672089, 61273274, 61572064, and National Key Technology R&D Program of China 2012BAH01F03.

References
[1] W. Tao, K. Li, and K. Sun, “SaCoseg: Object Cosegmentation by Shape Conformability,” Image Processing, IEEE Transactions on, 24(3), pp. 943-955, 2015.
[2] A. Joulin, F. Bach, and J. Ponce, “Multi-class cosegmentation,” In Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pp. 542-549, June 2012.
[3] C. Rother, T. Minka, A. Blake, and V. Kolmogorov, “Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrf,” In Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, Vol. 1, pp. 993-1000, June 2006.
[4] Y. Mu, and B. Zhou, “Co-segmentation of image pairs with quadratic global constraint in MRFs,” In Computer Vision–ACCV, Springer Berlin Heidelberg, pp. 837-846, 2007.
[5] L. Mukherjee, V. Singh, and C. R. Dyer, “Half-integrality based algorithms for cosegmentation of images,” In Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pp. 2028-2035, June 2009.
[6] D. S. Hochbaum, and V. Singh, “An efficient algorithm for co-segmentation,” In Computer Vision, IEEE 12th International Conference on, pp. 269-276, September 2009.
[7] S. Vicente, V. Kolmogorov, and C. Rother, “Cosegmentation revisited: Models and optimization,” In Computer Vision–ECCV, Springer Berlin Heidelberg, pp. 465-479, 2010.
[8] S. Vicente, C. Rother, and V. Kolmogorov, “Object cosegmentation,” In Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pp. 2217-2224, June 2011.
[9] K. Y. Chang, T. L. Liu, and S. H. Lai, “From co-saliency to co-segmentation: An efficient and fully unsupervised energy minimization model,” In Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pp. 2129-2136, June 2011.
[10] Y. Schnitman, Y. Caspi, D. Cohen-Or, and D. Lischinski, “Inducing semantic segmentation from an example,” In Computer Vision–ACCV, Springer Berlin Heidelberg, pp. 373-384, 2006.
[11] A. Joulin, F. Bach, and J. Ponce, “Discriminative clustering for image co-segmentation,” In Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pp. 1943-1950, June 2010.
[12] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, “icoseg: Interactive co-segmentation with intelligent scribble guidance,” In Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pp. 3169-3176, June 2010.
[13] M.W. Mahoney, L. Orecchia, and N. K. Vishnoi, “A local spectral method for graphs: With applications to improving graph partitions and exploring data graphs locally,” The Journal of Machine Learning Research, 13(1), pp. 2339-2365, 2012.
[14] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, 34(11), pp. 2274-2282, 2012.
[15] M. Varma, and A. Zisserman, “A statistical approach to texture classification from single images,” International Journal of Computer Vision, 62(1-2), pp. 61-81, 2005.
[16] J. Shi, and J. Malik, “Normalized cuts and image segmentation,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22(8), pp. 888-905, 2000.
[17] F. Meng, H. Li, G. Liu, and K. N. Ngan, “Object co-segmentation based on shortest path algorithm and saliency model,” Multimedia, IEEE Transactions on, 14(5), pp. 1429-1441, 2012.