Graph-Based Fast Image Segmentation*

Dongfeng Han, Wenhui Li, Xiaosuo Lu, Lin Li, and Yi Wang

College of Computer Science and Technology, Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education, Jilin University, Changchun, 130012, P.R. China
jlu_hdf@126.com

Abstract. In this paper, we describe a fast semi-automatic segmentation algorithm. A nodes aggregation method is proposed for improving the running time and a Graph-Cuts method is used to model the segmentation problem. The whole process is interactive. Once the users specify the interest regions by drawing a few lines, the segmentation process is reliably computed automatically no additional users' efforts are required. It is convenient and efficient in practical applications. Experiments are given and outputs are encouraging.

1 Introduction

Image segmentation is a process of grouping together neighboring pixel whose properties are coherent. It is an integral part of image processing applications like medical images analysis and photo editing. Many researchers are focus on this subject [1], [2], [3], [4], [5], [6], [7], [8], [9] and [10]. However, even to this day, many of the computational issues of perceptual grouping have remained unresolved. Semi-automatic segmentation techniques that allow solving moderate and hard segmentation tasks by modest effort on the part of the user are becoming more and more popular. In the analysis of the objects in images it is essential that we can distinguish between the objects of interest and the rest. This latter group is also referred to as the background. The techniques that are used to find the objects of interest are usually referred to as segmentation techniques: segmenting the foreground from background.

In this paper, we present an efficient method for semi-automated image segmentation for common images. Our method can exactly extract the interest objects such as people, animals and so on. This paper's contribution is twofold.

First, we introduce the nodes aggregation method to the image segmentation. It is a powerful tool for reducing running time.

Second, we propose a novel semi-automation segmentation scheme based on nodes aggregation and Graph-Cuts, which has several favorable properties:

1. Capable of solving interactive segmentation problem.
2. Performs multi-label image segmentation (the computation time does not depend on the number of labels).
3. Fast enough for practical application using nodes aggregation.

* This work has been supported by NSFC Project 60573182, 69883004 and 50338030.

4. It is extensible and allows constructing new families of segmentation algorithms with specific properties.

In section 2 describes our method in detail. Section 3 presents the experiments. Conclusions and future work are given in section 4.

2 Our Method

2.1 Overview

Just as mentioned in [7], the segmentation problems focus on two basic questions:

1. What is the criterion that one wants to segment?
2. Is there an efficient technique to carry out the segmentation?

In this paper, our goal is to give an answer about the two questions. There are two main contributions of our segmentation algorithm. We will introduce each of these ideas briefly below and then describe them in detail in subsequent sections. The first contribution of this paper is a new coarse image representation called nodes aggregation that allows for very fast completing segmentation. The second contribution of this paper is combining nodes aggregation with Graph-Cuts that can produce an elaborate segmentation result.

2.2 Graph-Cuts Based Image Segmentation

Inspired by [3], we use Graph-Cuts as energy minimization technique for the interactive segmentation. Furthermore, at the object marking step, we propose an efficient interactive segmentation algorithm by employing multi-scale nodes aggregation and Graph-Cuts.

We briefly introduce some of the basic terminology used throughout the paper. An image that contains $N = n \times n$ pixels, we construct a graph $G = (V, E, W)$ in which each node $i \in V$ represents a pixel and every two nodes i, j representing neighboring pixels are connected by an edge $e_{i,j} \in E$. Each edge has a weight $w_{i,j} \in W$ reflecting the contrast in the corresponding location in the image. We can connect each node to the four or eight neighbors of the respective pixel, producing a graph. The graph can be partitioned into two disjoint sets, “O”, “B” where $O \cup B = V$, $O \cap B = \emptyset$. These tasks can be viewed as labeling problems with label 1 representing object and 0 otherwise. Finding the most likely labeling translates to optimizing an energy function. In vision and image processing, these labels often tend to vary smoothly within the image, except at boundaries. Because a pixel always has the similar value with its neighbors, we can model the optimization problem as a MRF. In [3], the authors find the most likely labeling for some given data is equivalent to seeking the MAP (maximum a posteriori) estimate. A graph is constructed and the Potts Energy Model (1) is used as the minimization target.

$$H(G) = H_{data}^{(i)} + H_{smooth}^{(i,j)} \quad (1)$$

$i \in V$ $\{i,j\} \in N$

The graph G contains two kinds of vertices: p -vertices (pixels which are the sites in the associated MRF) and l -vertices (which coincide with the labels and will be terminals in the graph cut problem). All the edges present in the neighborhood system N are edges in G . These edges are called n -links. Edges between the p -vertices and the l -vertices called t -links are added to the graph. t -links are assigned weights based on the data term (first term in Equations 1 reflecting individual label-preferences of pixels based on observed intensity and pre-specified likelihood function) while n -links are assigned weights based on the interaction term (second term in Equation 1 encouraging spatial coherence by penalizing discontinuities between neighboring pixels). While n -links are bi-directional, t -links are un-directional, leaving the source and entering the sink. A cut segment the graph.

The object of segmentation is to minimize the energy function (1). The first term reflects individual label-preferences of pixels based on observed intensity and pre-specified likelihood function. The second term encourages spatial coherence by penalizing discontinuities between neighboring pixels. So our goal is minimize the energy function and make it adapt to human vision system. In [3] authors give the construction of the graph in detail. A different way is used to construct the graph in this paper because its construction way is so complex that in practice it will be inconvenient and it will influence the running time.

First a k-Means clustering method is applied on the seed region including “O” and “B”. We use the algorithm described in [11] which performs well in practice. (A C code can be found on his homepage) Then, for each node i , the minimum distance from its color $color(i)$ to foreground clusters is computed as:

$$d_i^O = \min_{j \in O} \|color(i) - kMeans_j\| \quad (2)$$

$$d_i^B = \min_{j \in B} \|color(i) - kMeans_j\| \quad (3)$$

So the energy of the first term is:

$$H_{data}(i=0) = \begin{cases} CONST & \text{if } i \in B \\ 0 & \text{if } i \in O \\ \frac{d_i^B}{d_i^B + d_i^O + \varepsilon} & \text{if } i \neq 1 \text{ or } 0 \end{cases} \quad (4)$$

$$H_{data}(i=1) = \begin{cases} 0 & \text{if } i \in B \\ CONST & \text{if } i \in O \\ \frac{d_i^O}{d_i^B + d_i^O + \varepsilon} & \text{if } i \neq 1 \text{ or } 0 \end{cases}$$

Where $i = 1$ or 0 means that this node belongs to Object or Background.

The second term $H_{\{i,j\} \in N}^{smooth}(i, j)$, which is define as follow:

$$H_{\{i,j\} \in N}^{smooth}(i, j) = \exp\left(\frac{-\|I(i) - I(j)\|^2}{\sigma I}\right) \times \exp\left(\frac{-\|L(i) - L(j)\|^2}{\sigma L}\right) \quad (5)$$

Where $I(i)$ is the color of the node i , $L(i)$ is the location of the i . So this term not only reflects the neighbor smoothness but also reflects the influence of two nodes due to distance.

So once a user specifies the seeds of object and background then a graph can be constructed as described above. Then using the Max-Flow algorithm (a more efficient implementation is described in [12]), we can efficient minimize the energy. The output is divided into two parts: one is object the other is background.

2.3 Nodes Aggregation

For a 512×512 pixel image, it will be slow to segment the whole graph straightly. We introduce a multi-scale nodes aggregation method to construct a pyramid structure over the image. Each procedure produces a coarser graph with about half size, and such that Graph-Cuts segmentation in the coarse graph can be used to compute precision segmentation in the fine graph. The coarsening procedure proceeds recursively as follows. The finest graph is denoted by $G^0 = (V^0, W^0, E^0)$ and a state vector $d = \{d_1, d_2, \dots, d_N\}$, where $N = \|V\|$ is defined:

$$d = \begin{cases} 1 & \text{if } i \in O \\ 0 & \text{if } i \in B \end{cases} \quad (6)$$

$d_i = 1$ means pixel i belonging to object, otherwise belonging to background. Supposing $G^s = (V^s, W^s, E^s)$ and d^s is defined at scale s . A set of coarse representative nodes $C \subseteq V^s = \{1, 2, 3, \dots, n\}$ and $\|V^s\| \leq N$ is chosen, so that every node in $V^{s-1} \setminus C$ is strongly connected to C . A node is strongly connected to C if the sum of its weights to node in C is significant proportion of its weights to nodes outside C . The principle can be found in [9]. Each node in C can be considered as representing an aggregation node. Thus we decompose the image into many aggregates. Then a coarse graph $G^{s-1} = (V^{s-1}, W^{s-1}, E^{s-1})$ and a state vector $d^{s-1} = \{d_1^{s-1}, d_2^{s-1}, \dots, d_{N^{s-1}}^{s-1}\}$ at scale $s-1$ are defined. The relationship between W^s and W^{s-1} , d^s and d^{s-1} are:

$$w_{k,l}^s = \sum p_{i,k}^{[s-1,s]} w_{i,j}^{s-1} p_{j,l}^{[s-1,s]} \quad (7)$$

It can be written as the matrix formulation:

$$W^s = P^{[s-1,s]T} W^{s-1} P^{[s-1,s]} \quad (8)$$

So relationship between d^s and d^{s-1} are:

$$d^{s-1} = P^{[s-1,s]} d^s \tag{9}$$

A more detail description can be found in [9]. So we should find P which is called the inter-scale interpolation matrix in [10]. Once P is found, the pyramid structure can be constructed recursively. The selection of P can be found in [8]. In the end a full pyramid has been constructed. The graph cuts method can be used to segment the coarse graph, then applying a top-down sharpening of the boundaries. In the end each represent node at coarser level is assigned to low level, which produces a nodes reassembling procedure. That means a node i originally belonging to a represent node j can reassembling to a different represent node k due to the weight change. So a node i change its original state value from background to object (supposing it originally belongs to background).

Fig. 1. Some test images and the segmentation results

3 Experiments

We first give the running time for aggregation segmentation and non-aggregation segmentation in Table 1. The test images are shown in Fig. 1 from (a) to (e). The time is the total running time from beginning to end. Because we use nodes aggregation to reduce the number of nodes, the speed can meet the real-time demands. There are some methods for nodes aggregation. We use a simply but efficient method. For more accurate result we can develop new methods for nodes aggregation. Also the ratio of running time is about 20 times faster than non-aggregation. In Fig. 1, some segmentation results are shown. For some hard border images such as Fig. 1 (a) (c), our method can exactly segment the images. For some soft border images such as Fig. 1 (b) (d), our method can also give satisfying results. The outputs are encouraging. We also apply nodes aggregation segmentation method to interest region-based image retrieval system and get a satisfying performance.

Table 1. The comparisons of running time between non-aggregation and aggregation

Images	Non-aggregation	Aggregation
(a)	20s	1.0s
(b)	29s	1.5s
(c)	35s	2.2s
(d)	32s	1.9s

4 Conclusions and Future Work

In this paper an efficient semi-automatic segmentation method is proposed. The main contribution of this paper is introducing nodes aggregation for improving running time which is very important in practical applications. In the future, automatic segmentation algorithm based on nodes aggregation and Graph-Cuts will be developed. We believe it will be a powerful tool in practical applications.

Acknowledgments

This work has been supported by NSFC Project 60573182, 69883004 and 50338030.

References

1. L. Reese.: Intelligent Paint: Region-Based Interactive Image Segmentation, Master's thesis. Department of Computer Science, Brigham Young University, Provo, UT. (1999)
2. E. N. Mortensen and W. A. Barrett.: Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60, 5. (1998) 349–384
3. Y. Boykov and M. Jolly.: Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. *Proc. IEEE Conf. Computer Vision*. (2001) 105–112
4. C. Rother, A. Blake and V. Kolmogorov.: Grabcut-interactive foreground extraction using iterated graph cuts. In *Proceedings of ACM SIGGRAPH*. (2004)

5. D. Comanicu, P. Meer.: Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, May. (2002)
6. J. Shi and J. Malik.: Normalized Cuts and Image Segmentation. *Proc. IEEE Conf. Computer Vision and Pattern Recognition.* (1997) 731-737
7. J. Shi and J. Malik.: Normalized cuts and image segmentation. *PAMI*, 22(8) (2000) 888-905
8. E. Sharon, A. Brandt, and R. Basri.: Fast multiscale image segmentation. *Proc. IEEE Conf. Computer Vision and Pattern Recognition.* (2000) 70-77
9. E. Sharon, A. Brandt, and R. Basri.: Segmentation and boundary detection using multiscale intensity measurements, *Proc. IEEE Conf. Computer Vision and Pattern Recognition.* (2001) 469-476
10. Meirav Galun, Eitan Sharon, Ronen Basri, Achi Brandt.: Texture Segmentation by Multiscale Aggregation of Filter Responses and Shape Elements, *Proc. IEEE Conf. Computer Vision.* (2003)
11. Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, Angela Y. Wu.: A Local Search Approximation Algorithm for k-Means Clustering, In *Proceedings of the 18th Annual ACM Symposium on Computational Geometry.* (2003)
12. Y. Boykov and V. Kolmogorov.: An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Computer Vision, *Proc. Int'l Workshop Energy Minimization Methods in Computer Vision and Pattern Recognition, Lecture Notes in Computer Science*, Sept. (2001) 359-374