Object Boundary Segmentation Using Graph Cuts Based Active Contours

† Ning Xu ‡ Ravi Bansal † Narendra Ahuja

†Beckman Institute and ECE Department
University of Illinois at Urbana-Champaign
405 N. Mathews Ave., Urbana, IL 61801
{ningxu,ahuja}@vision.ai.uiuc.edu

‡Imaging and Visualization Department,
Siemens Corporate Research
755 College Road East, Princeton, NJ 08540
ravi@scr.siemens.com

Abstract

In this paper we propose an iterative graph cuts based active contours approach to segment an object boundary out of background. Given an initial boundary nearby the object, the graph cuts based active contour can iteratively deform to the object boundary even if there are large discontinuities and noise. In each iteration, the area of interest is a certain neighborhood of the previously estimated boundary. The problem is formulated as a multi-source multi-sink $s-t$ minimum cut problem in that neighborhood and solved using node identification. The result of each step is globally optimal within the area of interest. On one hand, this area of interest enables the active contour to break away from the constraints of previously estimated boundary and isolated noise points in the image. On the other hand, our method generates multiple sources such that the result boundary should be long enough and hence overcomes the well-known shortcoming of the cost function of min-cut. It is shown that the proposed approach will converge to a final boundary. If this final boundary is not satisfactory, our approach is inherently suitable for interactive correction. As an advantage of using graph cuts, the proposed approach can be easily and seamlessly used to segment two dimensional(2D), three dimensional(3D) and possibly higher dimensional objects.

1. Introduction

Accurate extraction of object contours is an important problem in a wide range of image processing applications. The segmented object boundary can be used to recover shape descriptions which can then be further utilized for high–level vision tasks, such as tracking and shape analysis[9]. There are many problems in extracting accurate and optimal boundaries of the objects, such as discontinuities in object boundaries and noise in the image. The discontinuities may be very large and noise may be due to single noise points within the neighborhood or the whole data is corrupted by noise.

A number of methods have been proposed for extracting object boundaries both in the computer vision and medical image processing literature. Snakes [7], the active contour models, remain to be very effective in extracting object contours in images with large amount of degradations [3]. For segmenting an object from an image, Snakes depend upon external image pixel intensity derived forces (external energy) to pull the contours towards desired image features and its internal energy to find smooth boundaries. Implicit active shape models embed contours as the zero level set of a higher dimensional function and then solve the equation of motion described by a partial differential equation [8]. These methods are not very convenient for user interaction. Parametric models, where the deforming contour is described using few parameters [11], have been successfully used for the cases where the topology of the extracted surface or the contour is simple. However, these methods tend to converge to local optima. Dynamic programming based approaches have been proposed [2, 6] where the extracted contour is optimal. However, these methods do not scale properly from extracting contours to extracting surfaces.

Graph cuts have been applied as a global optimization method for image segmentation [14]. However, it suffers from a bias toward cuts with short boundaries and thus small regions due to the cost function of minimum cut. Some other approaches [10, 13] introduced different cost functions but they are either time consuming or can only be solved approximately. Also, these approaches are partitioning an image into different regions rather than extracting an accurate object boundary. Interactive graph cuts [4] introduced $s-t$ minimum cut as an optimization method with user interactively identified object and background regions. However, this approach depends a lot on the interactive input and can not break away from it. Authors in [12] integrate global segmentation and interactions to yield better results .

In this paper we present a graph cuts based active contour algorithm which iteratively applies multi-source multi-sink $s-t$ minimum cut to segment an object out of background.

Given an initialization of the object boundary, our active contour can automatically deform to the object boundary even if there are large object boundary discontinuities and noise in the image. Unlike parametric and level set based approaches, our estimated boundary is optimal. Unlike dynamic programming, our algorithm can easily and seamlessly be extended from estimating 1D contour to a 2D surface and higher dimensional manifolds. In each the iteration, area of interest is a certain neighborhood of the previous boundary and problem is formulated as a multi-source multi-sink $s-t$ minimum cut problem. Using a simple operation called *node identification*, the multi-source multi-sink $s-t$ minimum cut problem, in each iteration, is transformed into a single $s-t$ minimum cut problem. Note that as the result boundary should be long enough to include all these sources, our approach overcomes the shortcoming of the cost function of min-cut. The result of each step is globally optimal within the area of interest, which makes the active contour break away from the constraints of previous boundary and isolated noise points. It is also shown that our approach is guaranteed to converge and inherently suitable for user interaction to modify or guide the results, like snakes.

In the next section we describe our proposed approach in detail. Section 3 shows the results while extracting contours from 2D images and surfaces from 3D volume data set. Section 4 provides a discussion of our results and points towards future research directions.

2. Our Approach

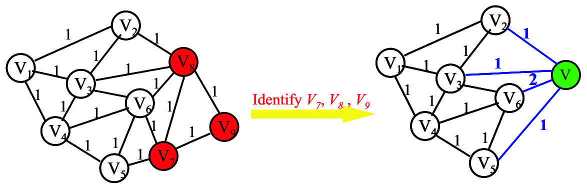


Figure 1: Node identification

2.1. Related graph theory

Graph-theoretic description of $s-t$ minimum cut can be found in many graph theory textbooks [1, 5]. The minimum cut considered in this paper is required to separate multiple source nodes from multiple sink nodes. A simple operation on a graph G called *node identification* identifies a set of nodes $\{V_1, V_2, ..., V_n\}$ into a single new node V, deleting self loops, if any, and merging parallel edges, as shown in Fig.1. For multi-source multi-sink $s-t$ minimum cut problem, we have the following theorem:

Theorem 1: (Minimum Cut with multiple sources and multiple sinks): The minimum cut of graph G which separate a source set $\{s_1, s_2, \ldots, s_n\}$ and a sink set $\{t_1, t_2, \ldots, t_m\}$ is exactly the $s-t$ minimum cut of the result graph after identifying $s_1, s_2, \ldots, s_n$ to a new source s and identifying $t_1, t_2, \ldots, t_m$ to a new sink t.

The proof follows as there is a one to one mapping between a cut (S, T) in the original graph G that separate $\{s_1, s_2, \ldots, s_n\}$ from $\{t_1, t_2, \ldots, t_m\}$ and $s-t$ cut (S', T') in the result graph G' after identifying the source set to s and the sink set to t, where $S = S' - s + \{s_1, s_2, \ldots, s_n\}$ and $T = T' - t + \{t_1, t_2, \ldots, t_m\}$. The capacities of the corresponding cuts are same since node identification only deletes self loops which will not be on the cuts. So if a cut (S', T') is a $s-t$ minimum cut in G', its corresponding cut (S, T) is a minimum cut in G that separate the source set and the sink set.

With this theorem, we can use $s-t$ minimum cut algorithms to solve the multi-source multi-sink minimum cut problem by simply identifying the multi-source as a single source and multi-sink as a single sink respectively.

2.2. Graph cuts based active contours

Given an initial boundary either by user interaction or automatic detection, our active contour algorithm iterates as follows:

1. Dilate the initial boundary into an area of interest with an inner boundary and an outer boundary, as shown in Fig.2.
2. Present the data within the area of interest using an adjacency graph.
3. Identify all the nodes on the inner boundary as a single source s and identifying all the nodes on the outer boundary as a single sink t.
4. Compute the $s-t$ minimum cut and the corresponding result boundary that separates the inner boundary from the outer boundary.
5. Repeat until the result converges.

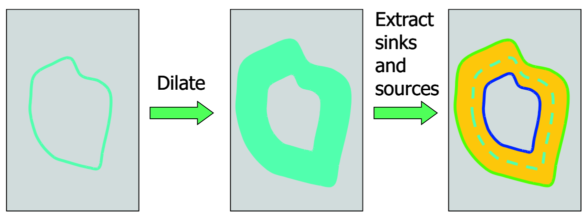


Figure 2: Using dilation to get sources, sinks and area of interest.

2.2.1 Dilation

Dilation used in step 1 has two advantages in each iteration. First, it generates an area of interest and the result boundary of current iteration is the globally optimal in this area. Since in each step the result boundary is better than previous one and is dilated to different area of interest in its next step, this makes the active contour to break away from the constraint of previous estimated result boundary and move towards a better result. Hence, our proposed graph cuts based active contour is insensitive to initialization. Second, dilation generates an inner boundary which is grouped as multiple sources in the corresponding graph. Since the min-cut cost function is prone to yield a small region, our approach of *node identification* overcomes this shortcoming as result boundary in each step should be large enough to include all these source nodes. Thus, our proposed approach can deform more than those active contours looking into only their connected neighborhood. Also, our method overcomes the shortcoming of the minimum cut cost function without having to use a cost function which is hard to solve.

2.2.2 Edge weights

Edge weights assignment is very important for graph theory based algorithms. Different methods to assign edge weights will yield different min–cut results. We have tried many different methods to assign edge weight $c(i,j)$. In this paper we are using $c(i,j) = (fg(i)+fg(j))^2$, where $fg(i) = \exp(-\mathrm{grad}(i)/\max_k(\mathrm{grad}(k)))$ and $\mathrm{grad}(i)$ is the image pixel intensity gradient at location i. This assignment method yields the best experimental results and intuitively, it will lead the active contour to the high gradient pixels.

2.2.3 Convergence

Theorem 2:(Convergence Theorem) Within finite data set, the graph cuts based active contour will either converge or oscillate between several results with same capacity after finite iterations.

Proof: Let C_i be the capacity of the result boundary R_i at the ith iteration, then $C_{i+1} \le C_i, i = 1,2,\ldots$. Finite data yields finite number of different results $R^u, 0 < u < N$. Since the dilation process and the edge weights are well defined, we should have $R_{i+k} = R_{j+k}$, for $k = 1,2,\ldots$ if $R_i = R_j$. If the algorithm doesn't converge, after N iterations, at least one result R^u will show up twice and then the sequence between this two R^u will oscillate. Also, since $C_{i+1} \le C_i, i = 1,2,...$, the capacity of each R_i within this sequence will be the same.

So if one boundary shows up twice in different iterations, we can terminate the algorithm.

2.2.4 Discontinuity and noise

If the object boundary is discontinuous, our approach will cut the object out with a path of smallest capacity, in most cases it is the shortest path, through the discontinuity area. Thus, the result boundary within discontinuity area is not necessary as the real object boundary, which could be any possible shape within this area. However, our approach provides an optimal result within its neighborhood based on the cost function. Also, if there is single noise point nearby the discontinuous boundary, our approach is able to jump that point which may have very high intensity gradient value. If the whole data is noisy, in our approach the boundary still cuts through the discontinuity area with an optimal path, though it may be zigzagged. All these properties are due to the fact that our result boundary is optimal within its neighborhood. In case our result boundary, though optimal in its neighborhood, is not what the user want, then our approach is inherently suitable for user to make interactive corrections.

2.3. Interactive corrections

Automatic segmentation approaches will sometimes lead to incorrect results. In our proposed method, if the segmentation result is not satisfactory, then the user can click a point on the result boundary, and our algorithm will automatically update the source set and sink set based on this input point. Three situations based on the user clicked points are distinguished.

1. the input point lies between the two boundaries.
2. the input point lies inside the inner boundary.
3. the input point lies outside of the outer boundary.

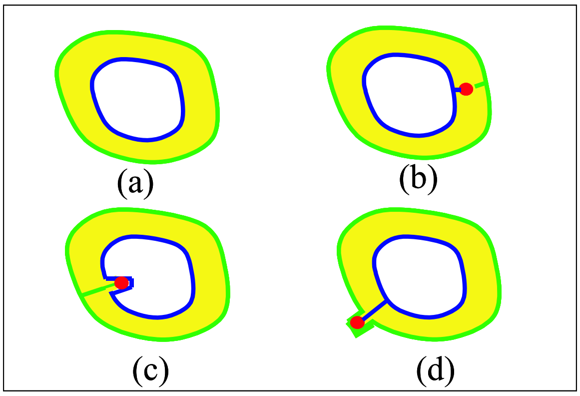


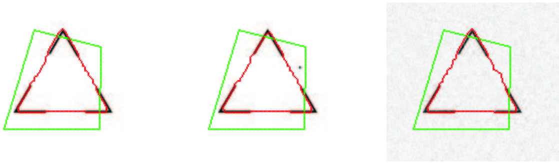
Figure 3: Source and sink set updating based on the input point. Yellow region is the area of interest, green indicates sources and blue indicates sinks.

These three situations are treated in different ways as shown in Fig.3. The main idea is that by updating the sources and sinks as indicated in Fig.3, the result should pass through the input point. In the following iterations, this user clicked point will lie between the two boundaries, and the final estimated boundary will pass through this point.

3. Experimental Results

Fig.4 shows that our approach is able to handle large discontinuities on the object boundary and noise in the image. Green polygon shows the initial boundary and the red contour shows the result boundary. Fig.4(a) is a synthetic image of a broken triangular shaped object. Fig.4(b) has an isolated noise point within the neighborhood of the discontinuity. While the original image is corrupted by Gaussian additive noise, the result is shown in Fig.4(c).

Fig.5 shows that our approach is insensitive to initial boundary. If the initial boundary is too small or too far



(a) (b) (c)

Figure 4: Experimental results

away, it is hard for our approach to find the real object boundary. However, interactive user corrections will be useful in these cases.

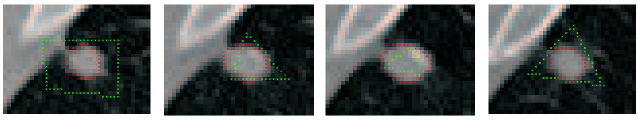
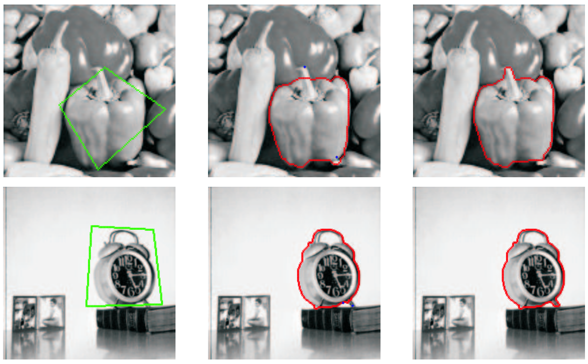


Figure 5: The approach is insensitive of initial boundary

Fig.6 shows that we can easily incorporate user interaction within our algorithm. Fig.6(a) is the original images and the initial boundaries. Fig.6(b) shows the estimated result boundary of our automatic approach as red curve. But it is not satisfactory and then the user inputs few(2) points, as marked with blue stars, to guide the active contour to the desired result. Fig.6(c) shows the final results.



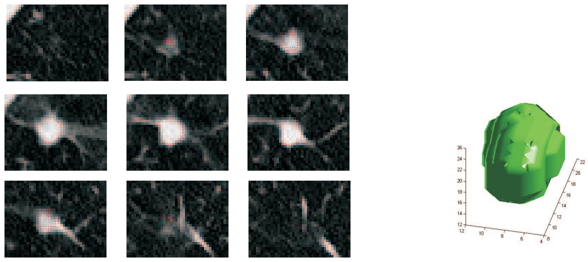
(a) (b) (c)

Figure 6: Experiments of interactive corrections

Fig.7 shows the application of our graph cuts based active surface for 3D lung nodule segmentation in CT volume data set. The nodule is attached to a few blood vessels. Our initialization here is a sphere nearby the object boundary. The estimated result surface is shown in each slice in Fig.7(a) and the reconstructed surface of the lung nodule is shown in Fig.7(b).

4. Discussion

In this paper we presented a graph cuts based active contour algorithm for object boundary segmentation. We first developed a method to transform a multi-source multi-sink minimum cut problem into a single $s-t$ minimum cut problem.



(a) (b)

Figure 7: 3D object segmentation

Based on this, using an iterative strategy, like the snakes, our approach is able to segment objects with large discontinuities in its boundary where there is no pixel intensity difference between the object pixels and the background. User interaction for possible modification can be seamlessly integrated into our algorithm. As a future work, we would like to incorporate the region information into our algorithm. Also, we want to extend our algorithm to segment multiple object from an image.

References

[1] Ravindra K. Ahuja, Thomas Magnanti, and James Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, 1993.

[2] Amir Amini et al. Using dynamic programming for solving variational problems in vision. *IEEE Trans. on PAMI*, 12(9), September 1990.

[3] A. Black and A. Yuille, editors. *Active Vision*. MIT Press, 1993.

[4] Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *The 8th ICCV*, 1:105–112, July 2001.

[5] Thomas Cormen, Charles Leiserson, and Ronald Rivest. *Introduction to Algorithms*. McGraw–Hill Companies, 1990.

[6] Davi Geiger, Alok Gupta, et al. Dynamic programming for detecting, tracking, and matching deformable contours. *IEEE Trans. on PAMI*, 17(3), March 1995.

[7] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988.

[8] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(2):158–175, 1995.

[9] C. Nastar and N. Ayache. Fast segmentation, tracking and analysis of deformable objects. *Fourth International Conference on Computer Vision*, pages 275–279, May 1993.

[10] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE CVPR*, June 1997.

[11] L. H. Staib and J. S. Duncan. Boundary finding with parametrically deformable models. , *IEEE Trans on PAMI*, 14(11):90–104, 1992.

[12] K. Tan and N. Ahuja. Selecting objects with freehand sketches. *The 8th ICCV*, 1:337–344, July 2001.

[13] S. Wang and J.M. Siskind. Image segmentation with minimum mean cut. *The 8th ICCV*, 1:517–524, July 2001.

[14] Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Trans. on PAMI*, 15(11):1101–1113, 1993.