33rd Annual International Conference of the IEEE EMBS
Boston, Massachusetts USA, August 30 - September 3, 2011

Biomedical Image Segmentation via Constrained Graph Cuts and Pre-segmentation

Zeyun Yu*, Ming Xu and Zhanheng Gao

Abstract— **In this paper we present a high-fidelity method for 2D and 3D image boundary segmentation. The algorithm is a novel combination of graph-cuts and initial image segmentation. The pre-segmentation using anisotropic vector diffusion and the fast marching method is employed so that the size of the graph being considered is significantly reduced. To further improve the segmentation accuracy, some user guidance is taken into account in finding the minimal graph cut. To this end, a user-friendly graphical user interface (GUI) is developed not only for visualization purposes but for user input and editing as well. The approaches and tools developed are validated on a number of 2D/3D biomedical imaging data, showing the high efficiency and effectiveness of our method.**

I. INTRODUCTION

THE boundary of an object is a generic feature defining the shape of the object. Boundary detection (or segmentation) in a digitized 2D or 3D image has been a fundamental research problem in image processing and informatics. Briefly speaking, boundary segmentation is a way to separate the features of interest from surrounding backgrounds or materials. Many other tasks, such as shape modeling and feature recognition, rely largely on correct segmentation. Manual segmentation is very tedious, inaccurate, and often subjective from person to person, even with the help of sophisticated graphical user interfaces [1-2]. It becomes even more challenging when a complicated 3D volume is under investigation. On the other hand, fully automated segmentation is still considered to be one of the hardest problems in the field of image processing, although various techniques have been described for automatic or semi-automatic segmentation. Commonly used methods include segmentation based on edge detection [3], region growing and/or region merging [4], active curve/surface motion [5], watershed immersion method [6], and normalized graph cut [7] and eigenvector analysis [8].

Among the existing computational approaches for segmentation, the deformable contour has drawn a lot of attention in recent years. There are two ways of representing deformable contours: one is parametric (i.e., the well-known snake model) [9] and the other is geometric (i.e., the level set method) [10]. The drawback of the parametric approach is that any topological change of contours would heavily complicate the procedures and data structures being used. By contrast, the level set method can handle topological changes naturally but is computationally more expensive due to the added dimension in the technique. When a single contour is considered in both parametric and geometric methods, one has to determine when the deforming process stops, which is often very tricky to do. In [4], a variant of the level set method was described to segment features in an image. This method, called the multi-seeded fast marching method, is used in conjunction with seed classification and region growing and merging [4, 11].

Another popular method used in image segmentation is the normalized graph cut method [7, 12]. In this approach, an image is treated as a graph, where each pixel corresponds to a node and an edge is created between two adjacent pixels. The segmentation result is obtained from the minimal graph cut, in a way similar to data clustering – the intra-class difference is minimized while the inter-class difference is maximized. Therefore, this method produces image segmentation in a more natural way. However, the original graph cut method is computationally expensive because the graph size generated from the original image may be huge, especially for 3D images. To remedy this problem, the concept of "superpixels" has been introduced to group a set of local, coherent pixels sharing similar texture or brightness into a region (or superpixel) [4, 13-14]. A simplified graph can be constructed and techniques such as graph-cuts or region-merging may be used to further group these regions into more meaningful segmentation. It has been estimated that using superpixels can tremendously reduce both time and memory consumptions [15].

While numerous segmentation algorithms have been explored, fully automatic segmentation methods with high accuracy still remains an open problem in medical image informatics, especially when the input image is corrupted by various noise or artifacts. Quite often, a doctor (or user) may need to edit the result after the segmentation is completed. This post-process, however, may be difficult for both the user and software developers when object shapes become complicated in 3D cases. Alternatively, one can ask the user to provide some guidance before the segmentation is executed. This pre-process is what we will adopt in our approach. To reduce the burden of the user, the guidance provided by the user must be as simple as possible for the user to input and for the program to understand. The easiest way is to pick a few points of interest (POI) and assign the type of each one of them. To this end, a user-friendly graphical interface is crucial and will be provided as part of the algorithms and toolkit described in the present paper.

Manuscript received March 26, 2011. This work is supported in part by an NIH Award (R15HL103497) from the National Heart, Lung, and Blood Institute (NHLBI) and by a subcontract from the National Biomedical Computation Resource (NIH P41 RR08605).

The authors are with the Department of Computer Science, University of Wisconsin-Milwaukee, Milwaukee, WI 53211, USA.

*Corresponding author: Z. Yu (phone: 414-229-2960; fax: 414-229-6958; email: yuz@uwm.edu).

II. METHODS

The segmentation method we shall describe below has three components: (1) initial segmentation (or so-called pre-segmentation) using anisotropic vector diffusion and the fast marching method; (2) superpixel-based, user-guided graph cut approach for final segmentation; and (3) a graphical user interface (GUI) that allows the user to guide the segmentation. The first two components are fully automatic and the third one only requires very minimal user input that can be easily provided through the GUI.

A. Initial Segmentation with Automatic Seed Selection and Fast Marching Method

While the level set method [10] shows a great generality in many applications, it has a major drawback due to its low computational efficiency. A significantly faster variant of the level set method is the fast marching method [16], designed for more specific problems where the speed function does not change signs. The traditional fast marching method is typically used for a single contour segmentation. In [4, 11], a multi-seeded strategy is utilized so that each seed initiates one independent contour. By this way, multiple contours competing with each other grow simultaneously and stop until all pixels have been conquered by one (and only one) of the contours. All such contours together produce a non-overlapping partition (or initial segmentation) of the original image. One of the keys in implementing this technique is how to automatically locate all the seed points that turn into the contours. To this end, we use gradient vector diffusion for two reasons. First, it can smooth the noise in a given vector field. This property is similar to image smoothing [17]. Second, vector diffusion can make non-zero vectors propagate toward the areas of zero-vectors. This property is extremely useful for many real-world images containing "flat" regions.

The initial gradient vectors are calculated using the central difference method, and then diffused by the following non-linear partial differential equations:

$$\begin{cases} \frac{du}{dt} = div(g(\alpha)\cdot\nabla u) \\ \frac{dv}{dt} = div(g(\alpha)\cdot\nabla v) \\ \frac{dw}{dt} = div(g(\alpha)\cdot\nabla w) \end{cases},$$

where (u, v, w) is the gradient vector in 3D space, $g(\ )$ is a decreasing function and α is the angle between the central vector and the surrounding vectors [11]. Once the gradient vector field is diffused, we detect the *source* points, where all the neighboring vectors point outwards. All the *source* points are regarded as the seed points in our algorithm (Fig. 1A). Finally, with the multi-seeded fast marching method, we can get the initial segmentation (Fig. 1C).

B. Region Merging Using Constrained Graph Cuts

The result in the initial step is an over-segmentation of the input image. The small regions need to be merged in order to produce more meaningful features in the image. In this paper we shall treat the initial segmentation as a weighted graph, where each region is regarded as a "superpixel" or a node in the graph and an edge is formed between any two adjacent regions [14]. In this study each edge weight is calculated by the intensity difference and average gradient magnitude on the common boundaries between two adjacent regions. Other measurements, such as the shape of a region or intensity variance within a region, may also be incorporated into the edge weight function [19].

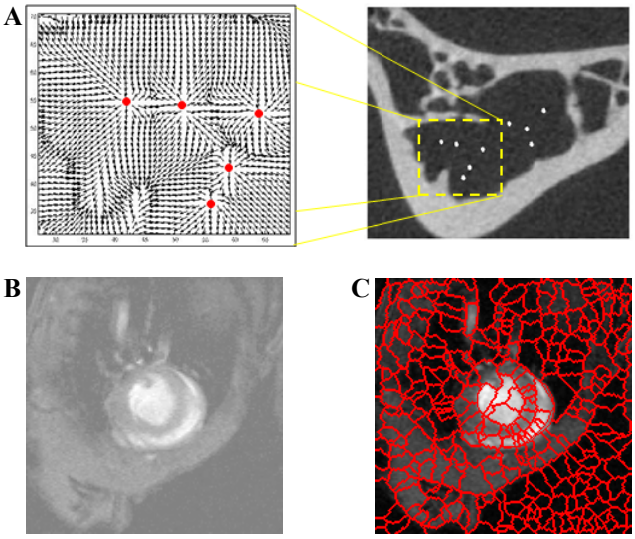

Figure 1 (A) An illustration of seed detection using the diffused gradient vector field. (B) An example of one cross section of the MRI imaging of the heart (image courtesy of Dr. Masahiko Hoshijima, UCSD). (C) Demonstration of the initial (over) segmentation.

Once a weighted graph G is generated, the normalized graph-cut based algorithm [7, 12] is employed to find the final segmentation. The normalized cut of a graph is defined as [7, 12]:

$$Ncut(A,B) = \sum_{a\in A, b\in B} w(a,b)\left(\frac{1}{\sum_{a\in A, v\in V} w(a,v)} + \frac{1}{\sum_{b\in B, v\in V} w(b,v)}\right),$$

where A and B are two subsets of the vertices V of the graph G and w(a, b) is the weight of the edge (a, b).

To further improve the segmentation accuracy, user guidance is incorporated into the original graph cut method. A number of seed points will be manually selected by the user using a graphical interface (see more details below). These seeds (and thus the corresponding regions in the initial segmentation or the superpixels in the graph) will be pre-classified into several groups, each corresponding to one type of features (or background). Since these seeds and their types are manually picked by the user, they must be kept in the final segmentations. In other words, the graph-cut algorithm becomes the following *constrained graph cut* (CGC) version:

Minimizing:

$$Ncut(A,B) = \sum_{a\in A, b\in B} w(a,b)\left(\frac{1}{\sum_{a\in A, v\in V} w(a,v)} + \frac{1}{\sum_{b\in B, v\in V} w(b,v)}\right),$$

$$\text{subject to } \begin{cases} s_i \in A, i = 1,2,\cdots,m \\ s_j \in B, j = 1,2,\cdots,n \end{cases}.$$

Here s_i and s_j are the seeds selected by the user for type A and B respectively. This definition can be extended to multiple components (types). Because the dimensionality of the input image (2D or 3D) has been handled during the initial segmentation, this problem is independent of the input image. For time efficiency, we adapt the iterative greedy method described in [20] to solve the above constrained graph-cut problem for image segmentation.

C. A Graphical User Interface for Seed Selection

We have designed a user-friendly graphical interface to facilitate the user input (i.e., selecting a couple of seeds and types by the user) to guide the constrained graph cut segmentation. The GUI and associated algorithms are written in a combination of C/C++ programs on top of the popular graphics library OpenGL, so that the software toolkit developed can be easily installed and executed in common systems. In particular, this toolkit provides a platform for users to select seed points for specific image features to be segmented. Fig. 2 shows an example of seed selection. The seeds can be picked in two ways: on 2D cross sections (in an arbitrary orientation) or 3D isosurfaces (see Window A in Fig. 2). The 3D coordinates and types assigned will be displayed in the property window and the user is allowed to change the types or delete the seeds (Window B). Other non-editable information will be displayed in Window C. Once the segmentation is finished, the results will be converted into high-quality surface and volumetric meshes for the downstream simulation problems [21]. This toolkit will be made available to the biomedical imaging and modeling community.

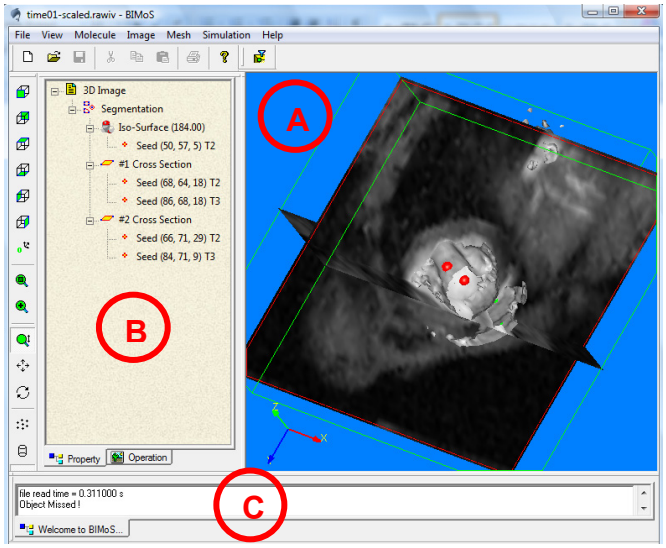

Figure 2 An illustration of our interactive toolkit. (A) The main display window is used to interactively manipulate the objects. Shown here are the cross sections and iso-surface of a 3D MRI image of the heart. (B) The editable property window shows a tree-like data structure with associated properties (i.e., the seed points and their types selected by the user). (C) The non-editable window shows some general information about the object as well as the operations (e.g., running time and size of the input image or mesh).

III. RESULTS

The algorithm described has been applied to a number of biomedical images in both 2D and 3D. Fig. 3A shows an example of electron microscopy image of cardiac sub-cellular structures, i.e., transverse tubules (t-tubule) surrounded by junctional sarcoplasmic reticulum (jSR). The size of the image is 273*316 pixels. Fig. 3B and 3C show the initial and final segmentation results, respectively, using our methods described above. From the results shown in Fig. 3D (segmentation with the original pixel-based method [20]) and Fig. 3E (segmentation with our method but without user guidance), we can see that the combined algorithm we proposed can achieve significantly better performance.

When the original image is corrupted with a high level of noise, it is often necessary to smooth the image before applying the segmentation algorithms. Fig. 4A shows an example of serial sectioning images of neurons (with image size of 463*322 pixels). Fig. 4B shows the segmentation result on the original image. By comparison, Fig. 4C and 4D show the smoothed image using the anisotropic filter [22] and the segmentation of the filtered image, respectively.

The algorithms described have been extended to segment 3D imaging data. Fig. 5 shows the segmented left ventricle from time-varying MRI images of the heart. One cross-section image is shown in Fig. 1B. Note that the surface models in Fig. 5 are reconstructed from the segmentation results by using the meshing toolkit as described in [21].

The comparisons of running time between our method and the pixel-based method [20] are given in Table 1. We can see that the total time for the two 2D examples are almost the same for both methods. However, the pre-segmentation in our method is executed only once in the beginning of the procedure. Once it is done, the user can run the constrained graph cut algorithm for as many as times in a very efficient way. Therefore, our method is very suitable for semi-automatic (interactive) image segmentation.

IV. CONCLUSION

This paper presents a new image segmentation method based on pre-segmentation and constrained graph cuts, which are encapsulated into a user-friendly GUI. Because of the super-pixels used, the proposed method shows considerably higher segmentation accuracy and speed (in case the user wants to repeatedly refine the segmentation), as compared to the original pixel-based method.

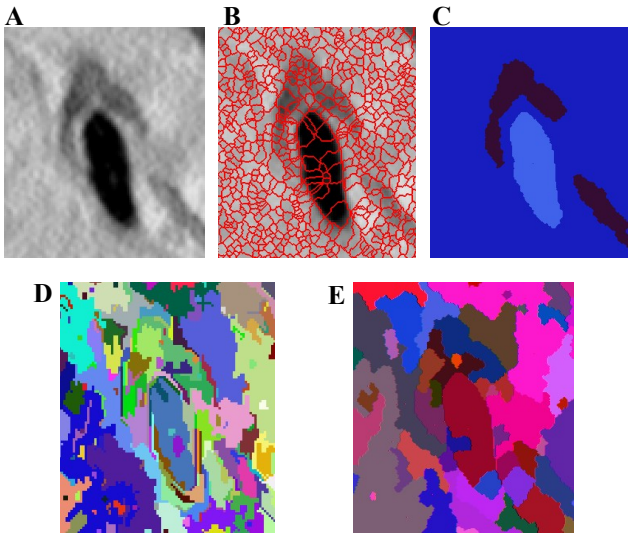

Figure 3 (A) The original electron microscopy image of cardiac sub-cellular structures. (B) Result of the pre-segmentation. (C) The image segmentation result using pre-segmentation and user-guided (or constrained) graph cut approach. (D) The segmentation result using the original pixel based method [20]. (E) The segmentation result with pre-segmentation but without user guidance.

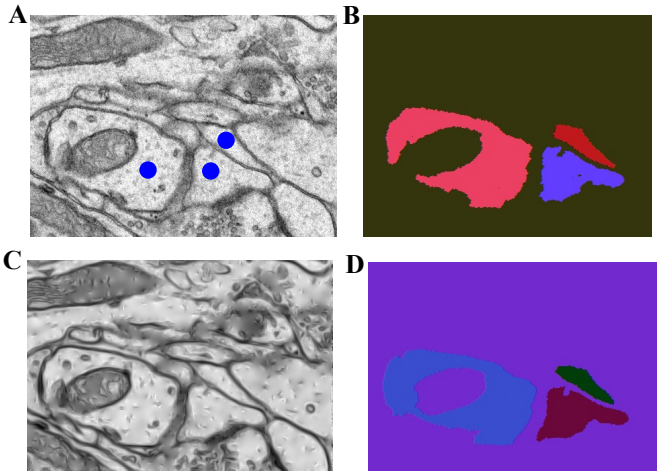

Figure 4 (A) Original serial sectioning image of neurons. The regions to be segmented are manually picked by the user as indicated by blue dots. (B) The segmentation of the image in (A). (C) The smoothed image with anisotropic filter. (D) The segmentation result on the filtered image in (C).

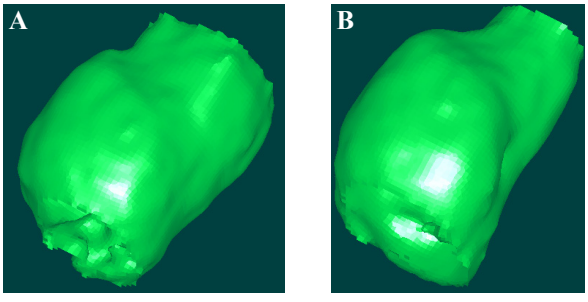

Figure 5 The surface models of the left ventricular chamber (see Fig. 1B for one of the cross section images) at two time steps. (A) Before the heart contraction (end-diastole). (B) 60ms after the heart contraction.

Table 1 Running time: our method vs. the pixel-based method [20]

Images	Sizes	Pixel-based graph-cuts	Our method	
			Pre-segmentation	Graph cuts
Fig. 3	273*316	2.14 s	1.94 s	0.16 s
Fig. 4	463*322	4.28 s	3.5 s	0.39 s

REFERENCES

[1] Li, Y., A. Leith, and J. Frank, *Tinkerbell - a tool for interactive segmentation of 3D data.* J. Struct. Biol., 1997. **120**: p. 266-275.

[2] Marko, M. and A. Leith, *Sterecon - three-dimensional reconstructions from stereoscopic contouring.* J. Struct. Biol., 1996. **116**: p. 93-98.

[3] Gonzalez, R.C. and R.E. Woods, *Digital image processing*. 1992: Addison-Wesley.

[4] Yu, Z. and C. Bajaj, *Image Segmentation Using Gradient Vector Diffusion and Region Merging*, in *Proceedings of International Conference on Pattern Recognition*. 2002. p. 941-944.

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

[6] Volkmann, N., *A novel three-dimensional variant of the watershed transform for segmentation of electron density maps.* J Struct Biol, 2002. **138**: p. 123-9.

[7] Shi, J. and J. Malik, *Normalized cuts and image segmentation.* Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1997: p. 731-737.

[8] Frangakis, A.S., et al., *Identification of macromolecular complexes in cryoelectron tomograms of phantom cells.* Proc Natl Acad Sci U S A, 2002. **99**(22): p. 14153-8.

[9] Kass, M., A. Witkin, and D. Terzopoulos, *Snakes: active contour models.* Int'l J. of Computer Vision, 1988. **1**: p. 321-331.

[10] Sethian, J.A., *Level Set Methods and Fast Marching Methods (2nd edition)*. 1999: Cambridge University Press.

[11] Yu, Z. and C. Bajaj, *Automatic Ultrastructure Segmentation of Reconstructed Cryo-EM Maps of Icosahedral Viruses.* IEEE Transactions on Image Processing, 2005. **14**(9): p. 1324-1337.

[12] Shi, J. and J. Malik, *Normalized cuts and image segmentation.* IEEE Trans. on Pattern Anal. Machine Intell., 2000. **22**(8): p. 888-905.

[13] Stein A., Hoiem D., and Hebert M., *Learning to Find Object Boundaries Using Motion Cues*. IEEE International Conference on Computer Vision (ICCV), October, 2007. p. 1-8.

[14] Ren X. and Malik J., *Learning a Classification Model for Segmentation* . In Proc. 9th Int. Conf. Comp. Vision, 2003. p. 10-17.

[15] Rohkohl, C. and Engel, K.: *Efficient Image Segmentation Using Pairwise Pixel Similarites*. In the 29th DAGM Symposium on Pattern Recognition, F.A. Hamprecht, C. Schnorr, and B. Jahne (Eds.), LNCS 4713, 2007. p. 254–263.

[16] Malladi, R. and J.A. Sethian, *A Real-Time Algorithm for Medical Shape Recovery.* Proceedings of International Conference on Computer Vision, 1998: p. 304-310.

[17] Perona, P. and J. Malik, *Scale-space and edge detection using anisotropic diffusion.* IEEE Trans. Pattern Anal. Machine Intell., 1990. **12**(7): p. 629-639.

[18] Yu, Z. and C. Bajaj, *Normalized gradient vector diffusion and image segmentation.* Proc. 7th European Conference on Computer Vision (ECCV'02), 2002. **3**: p. 517-530.

[19] Sumengen B., L. Bertelli, and B. S. Manjunath, *Fast and Adaptive Pairwise Similarities for Graph Cuts-based Image Segmentation*, Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop, 2006. p. 179-186.

[20] Felzenszwalb, P.F. and D.P. Huttenlocher, *Efficient Graph-Based Image Segmentation.* Int'l J. of Comput Vision, 2004. **59**: p. 167-181.

[21] Wang J. and Z. Yu, *Quality Mesh Smoothing via Local Surface Fitting and Optimum Projection*, Graphical Models, 2011. (In press)

[22] Fernandez J. and S. Li, *An Improved Algorithm for Anisotropic Nonlinear Diffusion for Denoising Cryo-tomograms*, J Struct Biol, 2003. **144**: p. 152-161.