

# Seed Extension Based Interactive Medical Volume Segmentation Method

Anjin Park\*, Hong-Lyel Jung, Joo Beom Eom, Jaesung Ahn and Byeong-Il Lee  
 Medical Photonics Research Center, Korea Photonics Technology Institute, Gwangju, Korea  
 \*Corresponding author

**Abstract**—This paper proposes an interactive segmentation method based on seed extension to tackle the problems of the min-cut/max-flows algorithm, which was extensively validated for many interactive segmentation applications. The extension is performed by constructing minimum spanning forests (MSF) from seed voxels imposed by users, which minimizes the weights of edges from the seeds to segmented lines (cuts). Compared with the graph cuts-based method, the proposed method segments the volume image into more than two regions of interests. Moreover, the proposed method performs 10 times faster when segmenting volumes composed of more than 240 slices, as the time complexity of constructing MSF is quasi-linear, whereas the min-cut/max-flow is polynomial.

**Keywords**—component; interactive segmentation; medical image processing; graph cuts; minimum spanning forests

## I. INTRODUCTION

Image acquisition devices, such as CT and MRI, produce 2D slices of specific areas of a scanned human body, allowing the medician to see inside the human body, and 3D volumetric images can be reconstructed by stacking up the 2D serially acquired slices. In biomedical imaging applications, image segmentation is typically used to locate objects of interest and their boundaries. The segmented results are used to visualize and analyze volumetric images of the regions, which are utilized as a way to make a diagnosis, fore examples, locating tumors and measuring tissues. Traditionally, the segmentation is performed by manual approach, which draws the boundaries of the regions slice by slice. This approach may take a lot of time, and may require expert knowledge to obtain accurate boundaries.

A number of computer-aided segmentation techniques have been proposed for medical images, and can be distinguished as automatic (unsupervised), interactive (semi-supervised), and supervised method [1]. The supervised method requires manually labelled training data for detecting specific objects in images and may limit the scope of these methods. Unsupervised method generated rough segmentation results, which can be further refined by the intervention of human experts [1]. Interactive segmentation [2,3] has become more and more popular in recent years, as the combination of human experts and machine intelligence can provide improved segmentation accuracy and efficiency with minimal user intervention [4].

The interactive method performs segmentation based on certain pixels (or voxel in 3D) that absolutely have to be part of the objects and certain pixels that have to be part of the background, which allows the user to get any desired segmentation results quickly via very intuitive interaction. The

use of graph cuts for segmentation of 3D surfaces has been extensively validated for medical volume images [5]. The graph cuts-based method can segment several types of images, as the method provides optimal solutions for the segmentation when the cost functions are clearly defined based on the user intervention [6]. However, since the running time of the min-cut/max-flow algorithm used to solve the solutions is slow when segmenting 3D volume images composed of tens or hundreds of slice, due to polynomial time complexity of the min-cut/max-flow algorithm [4]. Moreover, the graph cuts algorithm performs basically binary segmentation, which segments the input into two regions, foreground and background, and needs more times to get the segmentation results, as one-against-others approaches is used to segment more than two regions.

This paper proposes an interactive volume segmentation method based on minimum spanning forests (MSF) constructed from seed information marked by user. The goal of the proposed method segments the medical volume image into anatomical structures (segmentation target in this paper) as many as user wants and minimizes the running time, compared with the graph cuts-based method. The min-cut/max-flow algorithm of polynomial time complexity is not applicable for the volume images, as the number of voxels to be considered for the segmentation are exponentially increased according to the number of 2D slices. The proposed method uses Prim's algorithm that have quasi-linear time complexity to compute the MSF, and the greater the number of voxels is the relatively faster segmentation compared with the graph cuts-based method. In experiments, the proposed method shows 10 times faster than the graph cut-based method when the number of slices is 240, while shows 25 times faster when the number of slices 640. The proposed method showed improved results in terms of time performance, while the segmentation results were analogous to the results of the graph cuts-based method.

## II. MINIMUM SPANNING FOREST

We describe the basic terminology that pertains to minimum spanning forests in the context of the segmentation method.

In graph theory, a tree is an undirected graph in which any two vertices are connected by exactly one path. Let  $T$  be a subgraph of graph  $G = (V, E)$ . If  $T$  is a tree and a set of vertices of  $T$   $V(T)$  is same as a set of vertices of  $V(G)$ , then  $T$  is a spanning tree of  $G$ . A graph can have many different spanning trees. The minimum spanning tree (MST) is a spanning tree  $T' = (V, E')$  that minimizes  $f(T') = \sum_{e \in E'} w(e)$ , where  $w(e)$  is non-negative weight of each edge  $e \in E'$ . The weight of an edge represents the dissimilarity between the vertices connected by

the edge. This paper uses Prim's algorithm among many different algorithms, as the MST can be computed in quasi-linear time [8]. Table 1 shows algorithm constructing MST, and at the termination of the algorithm,  $(V, E_{new})$  is a MST on  $G$ .

TABLE I. PRIM'S ALGORITHM TO CONSTRUCT MST

Set $V_{new} = \{v\}$ , where $v$ is an arbitrary vertex in $V$ .
Set $E_{new} = \emptyset$ .
<b>while</b> $V_{new} \neq V$ <b>do</b>
Choose an edge $e_{u,v}$ with minimal weight such that $u$ is in $V_{new}$ and $v$ is not.
Add $v$ to $V_{new}$ and $e_{u,v}$ to $E_{new}$
<b>end</b>

MSF is a union of MST, and if the number of MST is  $n$ , the MSF is defined as follows.

$$V(F) = \bigcup_{i=1, \dots, n} V(T_i), E(F) = \bigcup_{i=1, \dots, n} E(T_i), \quad (1)$$

where  $V(X)$  and  $E(X)$  indicate a set of vertices and edges of subgraph  $X$ , respectively.

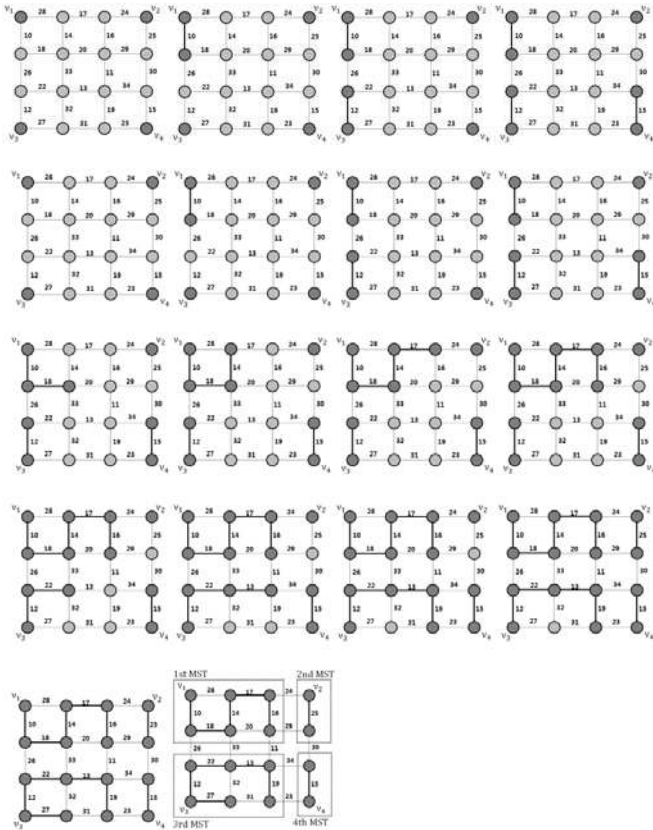


FIGURE I. EXAMPLE CONSTRUCTING MSF FROM SET OF INITIAL VERTICES

The problem of finding a relative minimum spanning forest is equivalent to the classical problem of finding a minimum spanning tree [9]. This provides a mean to derive an algorithm

for relative minimum spanning forests from any minimum spanning tree algorithm. One difference between the MST and MSF is that the initial number of vertex to construct the MST is one, while the initial number of vertex for the MSF can be more than one. The number of MST is determined by the number of initial vertices. Given the set of initial vertices  $V_{new} = \{v_1, \dots, v_n\} \in V(G)$ , where  $n$  is the number of vertices, i.e. MST, and  $1, \dots, n$  are indices of the MST, a vertex connected with edge  $e$  having a minimal weight among vertices adjacent to the initial vertices is added to  $V_{new}$  and the edge  $e$  is added to  $E_{new}$ . This process is repeated until all the vertex of the graph  $G$  are added to the  $V_{new}$ . Fig 1 shows the process constructing the MSF relative to the set of initial vertices  $V_{new} = \{v_1, v_2, v_3, v_4\} \in V(G)$ . If  $4 \times 4$  vertices are considered 2D image and the set of initial vertices are considered seeds, the 2D image can be segmented into multiple regions that the number of regions is same as the number of seed vertices.

### III. IMPLEMENTATION OF MSF FOR INTERACTIVE VOLUME SEMGNETATION

The proposed method performs image segmentation based on seeds. Here, the seed means approximate locations of the regions of interest (ex. anatomical structures). In this paper, three planes (horizontal, sagittal, and coronal planes) for the input volumetric image are displayed on the screen, as shown in figure 2(a), and the user marks seeds for the target regions by drawing lines or points on the 2D planes.

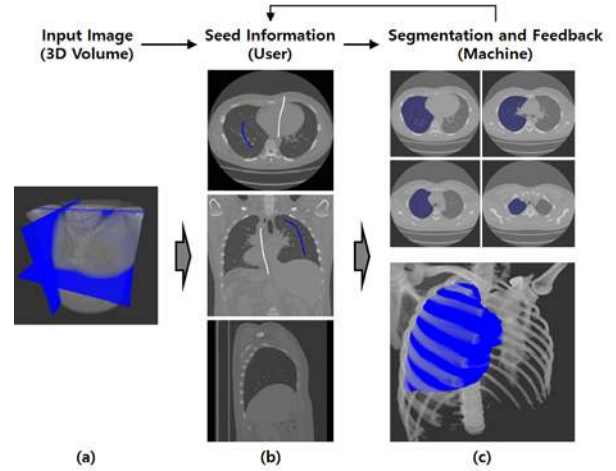


FIGURE II. FLOW OF INTERACTIVE VOLUME SEGMENTATION

Let the set of all the seeds marked by users as  $\mathcal{S} = \{\mathcal{S}_1, \dots, \mathcal{S}_{s_n}\}$  and the set of seeds for  $i^{\text{th}}$  region of interest as  $\mathcal{S}_i = \{s_1, \dots, s_{n_i}\}$ . Here, each element of  $\mathcal{S}_i$  indicates voxel,  $i$  indicates an index of the region,  $n_i$  indicates the number of voxels included in  $\mathcal{S}_i$ ,  $s_n$  indicates the number of regions to be segmented. Prim' algorithm computes the MSF by repeating the process of adding vertex connected to edge having minimal weight among the set of vertices adjacent to the set of vertices  $V_{new}$  to  $V_{new}$ . The proposed method performs image segmentation by repeating the process of adding voxel having minimal similarity among voxels adjacent to  $\mathcal{S}$  to  $\mathcal{S}$ . Here, the

adjacent voxel means that the voxel is a neighbor of one of voxels in  $\mathcal{S}$ , but is not included in  $\mathcal{S}$ , and is expressed as follows.

$$\mathbf{A} = \{s \notin \cup_{i=1}^{s_n} \mathbf{S}_i \mid N(s) \cap \cup_{i=1}^{s_n} \mathbf{S}_i \neq \emptyset\}, \quad (2)$$

where  $N(s)$  means the set of neighboring voxel of  $s$ , and six voxels on x-, y-, and z-axis are defined as the neighboring voxels.

To efficiently search the adjacent voxel connected with minimal edge, this paper uses a priority queue. The priority queue consists of more than one queue, where each queue additionally has a 'priority' associated with it. In a priority queue, a queue with low priority is served before a queue with high priority. The set of all the vertices  $\mathbf{A}$  adjacent to the set of initial vertices  $\mathcal{S}$  are added to the priority queue according to weight that is similarity between two voxels. For example, if the weight is 0, that is two voxels have same value, the voxel is added to the queue that the priority is 0, while if the weight is 255, the voxel is added to the queue that priority is 255. Based on the nature of the priority queue, the voxel  $v$  extracted from the lowest priority queue PQ is inserted to  $\mathcal{S}$ . For the next process, voxels adjacent to the  $v$  are added to the priority queue according to the weight, and the  $v$  is extracted from the queue. This process is repeated until all of the voxels of the input volume are inserted into  $\mathcal{S}$ .

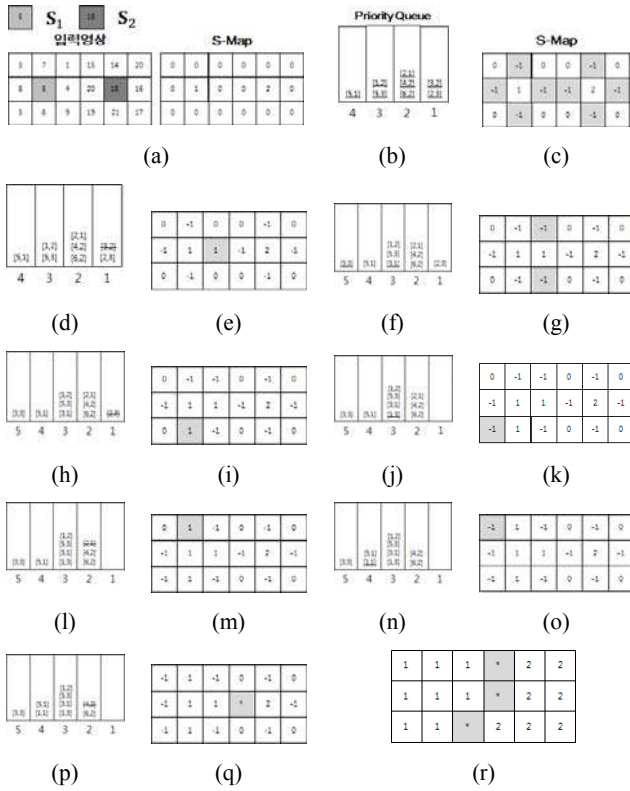


FIGURE III. STEPS TO SEGMENT IMAGE BASED ON MSF

Figure 3 shows steps to segment 2D image using the priority queue. Temporal image space, which has a same size as the input image, is created to save information generated during segmentation process corresponding to  $\mathcal{S}$ , and is called S-Map. The S-Map is initialized with the seed information, and corresponding segmentation index is set at coordinates of S-Map

corresponding to the seed voxels, while 0 is set at the other coordinates of the S-Map. Figure 3(a) shows an initial S-Map, and the segmentation index is set at coordinates of the set of initial seed voxels  $\mathcal{S} = \{\mathbf{S}_1, \mathbf{S}_2\}$ . The image segmentation is performed by setting segmentation index at coordinates having 0 value of the S-Map.

The process of the segmentation is as follows:

I. The set of voxel information adjacent to coordinates having segmentation index in the initial S-Map  $\{\{\mathbf{u}_1, \mathbf{v}_1, \delta_1, i_1\}, \dots, \{\mathbf{u}_n, \mathbf{v}_n, \delta_n, i_n\}\}$  are computed. Here,  $\mathbf{u}_j$  is a coordinate having 0 value,  $\mathbf{v}_j$  is a coordinate having segmentation index,  $i_j$  is a segmentation index, and  $\mathbf{u}_j \in N(\mathbf{v}_j)$ .  $\delta_j = |I_{\mathbf{u}_j} - I_{\mathbf{v}_j}|$  is a similarity between neighboring voxels  $\mathbf{u}_j$  and  $\mathbf{v}_j$ , calculated from an input image.

II. From the set of initial voxels, the initial priority queue is constructed by adding  $\mathbf{u}_j$  into queue that have a priority same as  $\delta_j$  (Figure 2(b)). To indicate that the coordinate  $\mathbf{u}_j$  is included in the priority queue, specific value (ex. -1) is assigned to the coordinate  $\mathbf{u}_j$  of the S-Map (Figure 2(c)). This means that the coordinate  $\mathbf{u}_j$  is included in the priority queue, but not assigned the segmentation index.

III. The coordinate of voxel  $\mathbf{u}_q$  are extracted from the lowest priority queue (Figure 2(d)), and the values of voxels adjacent to  $\mathbf{u}_q$  are checked in S-Map.

A. If the number of adjacent voxels that have more than 0 is only one, corresponding segmentation index is set at the coordinate  $\mathbf{u}_q$  of the S-Map (Figure 2(e)), and the voxel coordinate  $\mathbf{u}_j$  that the  $S_{\mathbf{u}_j}$  is 0 in the set of neighborhood voxels are added in the queue that that priority is  $\delta_j$  (Figure 2(f)). The specific value (ex. -1) are assigned at the coordinate  $\mathbf{u}_j$  of the S-Map (Figure 2(g)).

B. If the number of indices that are more than 0 are more than two, the corresponding voxel should be segmented line, as two segmentation indices are met at the voxel, and specific value (ex. \*) is assigned corresponding voxel in S-Map (Fig 2(q)).

IV. This process is repeated until segmentation index or segmented line is assigned all the voxels of the S-Map.

The final S-Map (Figure 3(r)) at the termination of the process is the segmentation result. In Fig 3, 2D image is used to easily explain the proposed method, but the proposed method can be extended to the 3D volume space without additional computation by considering Z-axis when obtaining neighboring voxels.

#### IV. EXPERIMENTAL RESULTS

Figure 4 shows a result image of the proposed method. Figure 4(a) shows seed information for segmentation target (humerus, radius, ulna, background), Figure 4(b) shows a segmented result, which is displayed on the sagittal plane, and Figure 4(c) shows a volume-rendered result. Figure 5 shows a result displayed on the (a) horizontal, (b) sagittal, and (c) coronal planes and (d) volume rendered result. As shown in Figures 4

and 5, the proposed method can segment more than two regions, and the number of regions is determined by the seed information.

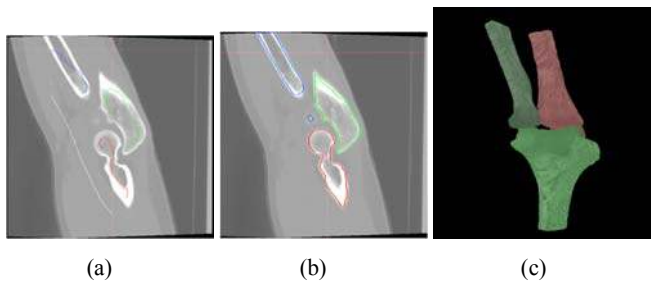


FIGURE IV. EXAMPLE OF SEGMENTATION RESULT OF PROPOSED METHOD: (A) SEED INFORMATION, (B) SEGMENTATION RESULT DISPLAY ON SAGITTAL PLANE, AND (C) VOLUME IMAGE OF SEGMENTED REGION

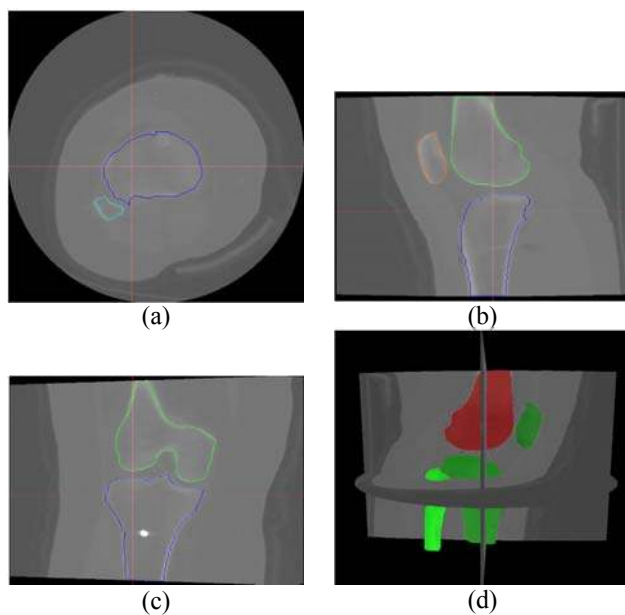


FIGURE V. EXAMPLE OF SEGMENTATION RESULT OF PROPOSED METHOD: (A) ON HORIZONTAL PLANE, (B) ON SAGITTAL PLANE, (C) ON CORONAL PLANE, AND (D) VOLUME IMAGE OF SEGMENTED REGION

Figure 6 shows a running time of the proposed method and graph cuts-based method according to the number of slices, with an assumption that the size of 2D slice is 512x512. To compare the proposed method with the graph cuts-based method, we tested segmenting two regions, background and foreground, and the graph cuts were implemented with source codes opened from the paper [6, 7]. As shown in Figure 6, the running time of the graph cuts-based method exponentially increases according to the number of slices, while the running time of the proposed method linearly increases proportional to the number of slices. The reason why the running time of the proposed method increases linearly is that time complexity of the Prim's algorithm used to compute the MSF  $O(E \log V)$ , where  $E$  is the number of edges and  $V$  is the number of voxels. On the other hands, the time complexity of the graph cuts is  $O(E^2V)$ , and the running time is proportional to the square of the number of edges multiplied by the number of voxels. Here, the number of edges

means the number of weights (similarities) to be considered for the segmentation.

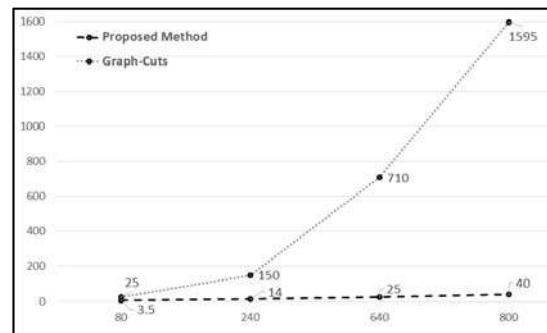


FIGURE VI. RUNNING TIME OF GRAPH CUTS-BASED AND PROPOSED METHODS

## V. CONCLUSIONS

This paper proposed an interactive volume segmentation method using seed extension to tackle the graph cuts based method. Based on the minimum spanning forests used to extend seeds, the proposed method could simultaneously segment more than two anatomical structures from one volume image. Moreover, the proposed method could segment high-quality CT and MRI volume images, which have many slices, more quickly compared with the previous methods.

## ACKNOWLEDGMENT

This work was supported by the ICT R&D program of MSIP/IITP. [B0101-15-0263, Development of Human Digital Multimedia Contents Generation and Service Technology]

## REFERENCES

- [1] F. Zhao and X. Xie, "Overview of Interactive Medical Segmentation," *Annals of the BMVA*, vol. 2013, no. 7, pp. 1-22, 2013.
- [2] Y. Boykov and G. Funka-Lea, "Graph Cuts and Efficient N-D Image Segmentation," *International Journal of Computer Vision*, vol. 70, issue 2, pp. 109-131, 2006.
- [3] X. Yan, N. Zhang, M.X. Nguyen, and B. Chen, "Volume Cutout," *Visual Computer*, vol. 21, pp. 745-754, 2005.
- [4] N. Lee, J. Caban, S. Ebasdollahi, and A. Laine, "Interactive Segmentation in Multimodal Medical Imagery using a Bayesian Transductive Learning Approach," *Proceedings of Medical Imaging: Computer-Aided Diagnosis*, pp. 72601W-1-10, 2009.
- [5] K. Li, X. Wu, Z. Chen, and M. Sonka, "Optimal Surface Segmentation in Volumetric Images – a Graph-theoretic Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 119-134, 2006.
- [6] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximation Energy Minimization via Graph Cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222-1239, 2001.
- [7] Y. Boykov and V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124-1137, 2004.
- [8] L. Vicent, "Morphological Grayscale Reconstruction in Image Analysis: Applicationis and Efficient Algorithms," *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 176-201, 1993.
- [9] R.L. Graham and P. Hell, "On the History of the Minimum Spanning Tree Program," *Annals of the History of Computing*, vol. 7, pp. 43-57, 1985