

Interactive Segmentation based on the Live Wire for 3D CT Chest Image Analysis

Kongkuo Lu and William E. Higgins*

Dept. of Electrical Engineering, Penn State University, University Park, PA 16802 USA

*Corresponding author. Fax: 1-814-863-5341. Email address: weh2@psu.edu

Abstract

The definition of regions of interest (ROIs) such as suspect cancer nodules or lymph nodes in 3D MDCT chest images is often difficult because of the complexity of the phenomena that give rise to them. Manual slice tracing has been used widely for years for such problems, because it is easy to implement and guaranteed to work. But manual slice tracing is extremely time consuming, subject to operator biases, and does not enable reproducible results. Numerous automated 3D image-segmentation methods have been proposed. But they are generally strongly application dependent, and even the most robust methods have difficulty in defining complex anatomical ROIs. To address this problem, the semi-automatic interactive paradigm, referred to as the live wire, has been proposed by researchers. In live-wire segmentation, the human operator interactively defines an ROI's boundary, guided by an active automated method. We propose 2D and 3D live-wire methods that improve upon previously proposed techniques. The 2D method incorporates an improved cost function to increase robustness and a search region to improve computational efficiency. For the new 3D method, the operator need only consider a few 2D image sections to begin and then an automated procedure defines the remainder of a 3D ROI's boundary. Tests run with five human operators on both 2D and 3D ROIs in 3D MDCT chest images demonstrate the exceptional performance of the methods. In particular, for all ROIs, the reproducibility was $> 97\%$ and the ground-truth correspondence was at least 97% . In addition, the 2D live-wire approach was at least 14 times faster than manual slice tracing for defining 2D ROIs, while the 3D method was at least 28 times faster than manual slice tracing for defining 3D ROIs. Finally, we describe a computer-based tool incorporating the methods and give an application of the tool for 3D MDCT-based planning and follow-on live guidance of bronchoscopy.

Keywords: Image segmentation, 3D imaging, computer-assisted image analysis, lung cancer, pulmonary imaging, interactive segmentation, live wire, image-guided surgery

Abstract word count: 307; Text word count: 7972
14 Figures, 10 Tables, 43 references.

1 Introduction

The definition of regions of interest (ROIs) in three-dimensional (3D) multi-detector CT (MDCT) chest images, is an important task for lung-cancer assessment [1–3]. Reliable automated methods exist, for example, for extracting the airway tree from 3D MDCT images [4,5]. But researchers have not been as successful at devising robust automated methods for extracting important diagnostic soft-tissue structures, such as suspect lung nodules and mediastinal lymph nodes [6,7].

Such structures often exhibit weak boundaries and can have greatly varying shape, size, inten-

sity, and 3D location. In addition, the quality of the image data can vary, depending on the MDCT scanner and scan conditions. Yet, it is vital that these structures be reliably extracted from given 3D MDCT images if 3D images are to be used successfully in a real clinical situation. This has often forced researchers to resort to tedious manual slice tracing to define ROIs, whereby a physician or technician manually traces a 3D ROI on each 2D section of a given 3D image. This was done, for example, in the efforts of [7, 8] for the central chest lymph-nodes.

Manual slice tracing has been used widely for years for such problems, because it is guaranteed to work and easy to implement. Yet, manual slice tracing is extremely time consuming — especially for high-resolution 3D MDCT images which may have hundreds of slices. It is also subject to operator biases (interobserver variability) and does not enable easily reproduced results on repeat trials (intraobserver variability) [9].

To counterbalance the drawbacks of manual interaction, researchers have devised semi-automatic methods. Such methods introduce operator control, which helps with robustness and accuracy, while still using automated analysis, which reduces human interaction time. One popular approach involves active contour models (or snakes) [10–13]. After given an initial approximation to an ROI’s boundary, a snake iteratively minimizes a cost function to settle upon a desired “optimal” contour. Performance relies partly on the quality of the initial boundary, which is either defined by the operator or defined based on a priori knowledge. Performance also relies, however, on the quality of the input image data, which often introduces unpredictable imperfections. Because the operator is only involved at the beginning of the process and must wait until an entire optimized contour is reached, the resulting boundary can often be inadequate. If the contour is unacceptable, then the whole process must be repeated — this is especially problematic for 3D ROIs. In addition, if part of an ROI’s boundary is weak, there is no guarantee that a snake will ever give a correct result.

The paradigm referred to as live wire, or intelligent scissors, arguably provides the best balance between manual interaction and automated analysis for general medical image segmentation [14–29]. A popular 2D live-wire implementation by Mortensen appears as the magnetic lasso tool in Adobe Photoshop [30]. The live wire, related to active contour analysis, casts the image segmentation problem as an optimal graph search, whereby a cost function is minimized via dynamic programming. The basic idea of live-wire image segmentation is as follows. The image grid is first cast as a directed weighted graph about a starting point (a seed) selected by the operator. Next, the

operator freely moves the mouse over the image. While this is done, an automated process finds the optimal-cost path between the first seed and the current mouse position (new candidate seed) — this contour is presented to the operator on the computer screen. The operator, who directly steers the boundary search, can either select the suggested contour or override it with a manual choice. In this manner, as illustrated in Figure 1 discussed later, the operator can rapidly piece together an ROI contour. The contribution of the live-wire paradigm is that it provides maximal human involvement, while providing near real-time interaction, to guarantee an accurate result in essentially any imaging circumstance.

Regarding past work on 3D live-wire segmentation, Schenk *et al.* employed a combination of the live wire and shape-based interpolation to find the contours of a 3D ROI [31]. The work, however, was not extended to 3D in a full sense, as it was only applied to selected sections and primarily relied on the interpolation technique. Falcão *et al.* formulated the first true 3D live-wire approach with applications to MRI-based bone segmentation [18, 21]. The method, however, requires significant user interaction, and can be difficult to implement. König *et al.* applied Mortensen’s 2D live-wire work to a 3D surface, but their method focused on assembling surface parts and performance depended greatly on 3D oversegmentation results [32, 33]. The approach of Salah *et al.* used the 2D live wire to propagate seeds to neighboring sections. But since the approach only searches around a seed’s 8-neighbors to locate a point on a new section, it cannot guarantee accuracy if contours change greatly from one section to the next or if the image data is noisy [34]. A preliminary 3D live-wire method of Hamarneh *et al.*, motivated by Mortensen’s 2D live-wire work, proposed a new seed-ordering method so that pre-defined seeds can be connected as closed boundaries automatically [35]. For convex regions, however, the method is sensitive to the selection of orthogonal slices and can give inaccurate results. The method of Souza *et al.* begins with a 2D boundary defined using the 2D live wire and then iteratively adjusts the boundary on subsequent adjacent slices [36]. This process continues until the entire 3D ROI is segmented. It is difficult, however, to define stopping constraints, and the method offers no guarantee of producing accurate results.

We propose new live-wire based methods for both 2D and 3D medical image segmentation. Our primary interest is in devising rapid fail-safe methods for defining important diagnostic structures, such as 3D pulmonary nodules and mediastinal lymph nodes, as they appear in 3D MDCT chest

images. The 2D method, which improves upon the method of Mortensen [16, 20], introduces an improved cost function to increase robustness; it also incorporates a new working region to improve computational efficiency. We next propose a new 3D live-wire method, which incorporates the 2D method. In the 3D method, the operator need only consider a few 2D image sections to begin and then an automated procedure defines the remainder of a 3D ROI’s boundary. Finally, we describe an interactive tool that incorporates these methods. The tool constitutes part of a large system we have been devising for 3D chest image assessment and for planning and guiding bronchoscopy [37, 38].

2 Methods

Section 2.1 first describes the improved 2D live-wire method. Next, Section 2.2 discusses the new 3D live-wire method, which draws upon the 2D method of Section 2.1. Finally, Section 2.3 presents the interactive tool that incorporates all methods.

2.1 Improved 2D Live-Wire Method

The improved 2D live-wire method builds off of the basic approach of Mortensen and Barrett [16, 20]. We introduce a new cost function that is more robust for weak boundaries. Also, at the beginning of the procedure, we enable the user to define a 2D rectangular working area. The working area helps reduce the time for computing the initial cost graph, restricts the subsequent graph search, and is suitable for diagnostically significant chest ROIs that generally have a limited size. With the 2D image of interest displayed on the computer monitor, the 2D live-wire method proceeds as follows:

1. The user defines a 2D rectangular working region that liberally encloses the ROI (Figure 1a).
2. The user moves the mouse over the first desired boundary point and selects it by a single left-mouse button click. This establishes the first boundary *seed* (Figure 1b). An automatically computed cost graph then establishes optimal paths from seed pixel p to all pixels within the working region. Optimality is based on the local-cost function $l(p, r)$ given by

$$l(p, r) = w_G f_G(r) + w_Z f_Z(r) + w_{D_1} f_{D_1}(p, r) + w_{D_2} f_{D_2}(p, r), \quad (1)$$

where $r \in N(p)$, $N(p)$ is the set of 8-neighbors of seed p , $f_Z(r)$, $f_G(r)$, $f_{D_1}(p, r)$, and $f_{D_2}(p, r)$

are cost components, and w_G , w_Z , w_{D_1} and w_{D_2} are user-specified weights. Figure 2 gives the automatic graph-search algorithm. Both the cost function (1) and graph-search algorithm are discussed further below.

3. The user moves the mouse along the desired contour. While doing this, an optimal cost path — a *live wire* — suggested by the internal graph search algorithm, automatically appears on the screen and “snaps on” to the optimal boundary between the current seed pixel p and the pixel that the mouse cursor currently rests over (Figure 1b). This occurs interactively as the user moves the mouse over the screen. The user can freely move the mouse over the screen and observe the contour being automatically suggested by the live wire.
4. When the suggested path is satisfactory, the user performs a single left-mouse button click to establish this portion of the ROI boundary. This also establishes a new seed pixel p .
5. The user repeats steps (3-4), with the ROI boundary being pieced together as new seeds are selected (Figure 1c). At any point, the user can double-click the left mouse button to complete a closed 2D ROI contour: the last selected seed and the first seed of step 2 are automatically connected by the optimal path found by the graph search (Figure 1d).

FIGURES 1 and 2 here.

The graph-search of Figure 2 uses Dijkstra’s graph-search algorithm for interactively suggesting locally optimal paths between the current seed pixel and the pixel pointed to by the mouse cursor in step 3 [19,20,39]. In order to convert the boundary-detection problem into a graph search, the image pixels within the selected working area are recast into a weighted directed graph. In the graph, the current seed pixel p serves as the root node, and edges emanating from root node connect to the seed’s 8-neighbor pixels $N(p)$. These 8-neighbors then serve as the next level of graph nodes. The 8-neighbors of these nodes are then connected to their respective 8-neighbors via edges, and the process is repeated for all pixels in the working area.

When handling a pixel $r \in N(p)$, if r is on the active list L and if $gtemp < g(r)$, our revised algorithm keeps r on L after updating its cost $g(r)$ and back pointer $pt(r)$, instead of removing r from the active list. We then requeue pixel r onto the active list when processing another pixel $q \neq p$ of L , where $r \in N(q)$. If r was instead removed from the active list at step 8 of Figure 2 as done in [20], then the active list must be completely resorted. Sorting is expensive when the active

list L is long; keeping r on L only requires partial resorting.

The local-cost function $l(p, r)$ in (1) consists of four separate cost components: $f_G(r)$ and $f_Z(r)$ are gradient-related costs, while $f_{D_1}(p, r)$ and $f_{D_2}(p, r)$ are costs related to gradient direction. The costs, $f_G(r)$ and $f_Z(r)$, are small when neighboring pixels r are on or close to region boundaries, while $f_{D_1}(p, r)$ and $f_{D_2}(p, r)$ are small when the two pixels, p and r , have similar gradient directions. The components $f_Z(r)$, $f_G(r)$, and $f_{D_1}(p, r)$ were suggested by Mortensen and Barrett, while $f_{D_2}(p, r)$ is a new cost [16, 20]. We briefly overview these costs below.

Let I be the input image, and define $I_x = \frac{\partial I}{\partial x}$ and $I_y = \frac{\partial I}{\partial y}$ as the horizontal and vertical components of the gray-scale gradient of I . Then, $f_G(r)$ is a gradient-magnitude cost given by

$$f_G(r) = 1 - \frac{G(r)}{\max(G)} \quad (2)$$

where the gradient magnitude G is approximated by $G = \sqrt{I_x^2 + I_y^2}$, G is weighted by $\sqrt{2}$ if r is a diagonal neighbor to p , and $\max(G)$ is the largest value of G over pixels within the selected working area. Thus, a pixel near a strong edge has a small f_G , while a pixel near a weak edge has a large f_G . Next, $f_Z(r)$ is a Laplacian zero-crossing cost denoted by

$$f_Z(r) = \begin{cases} 0, & \text{if } I_L(r) \leq T_{lap} \\ 1, & \text{if } I_L(r) > T_{lap} \end{cases} \quad (3)$$

where $I_L(r)$ is the normalized Laplacian of the original image at pixel r and T_{lap} is a threshold [40]. The Laplacian, a second-derivative operator, in principle changes sign across a region boundary pixel r . Thus, $f_Z(r) = 0$ ideally when a zero crossing exists at r ($I_L(r) = 0$) and 1 otherwise. But image imperfections make it difficult for this to occur exactly. Hence, the introduction of the threshold T_{lap} .

Since the edge information used by f_G and f_Z is often weak, it is unlikely that the desired ROI boundary can be found based only on gradient-related costs. The gradient-direction costs, $f_{D_1}(p, r)$ and $f_{D_2}(p, r)$, measure the similarity of gradient features between pixel p and its neighbor r , as shown in Figure 3. Let $D(p)$ be a unit vector representing the gradient direction at a pixel p , and $D'(p)$ be a unit vector perpendicular (rotated 90 degrees clockwise) to $D(p)$:

$$\begin{aligned} D(p) &= [I_x(p), I_y(p)] \\ D'(p) &= [I_y(p), -I_x(p)] \end{aligned}$$

The gradient-direction costs are calculated as follows:

$$f_{D_1}(p, r) = \frac{2}{3\pi} [\theta_1 + \theta_2] \quad (4)$$

$$f_{D_2}(p, r) = \frac{1}{\pi} |\theta_1 - \theta_2| \quad (5)$$

where

$$\theta_1 = \cos^{-1}[d_p(p, r)] \quad (6)$$

$$\theta_2 = \cos^{-1}[d_r(p, r)] \quad (7)$$

are angles defined between pixels (p, r) and vectors $(D'(p), D'(r))$, and

$$d_p(p, r) = D'(p) \cdot L(p, r)$$

$$d_r(p, r) = D'(r) \cdot L(p, r)$$

are vector dot products where

$$L(p, r) = \frac{1}{\|p - r\|} \begin{cases} r - p, & \text{if } D'(p) \cdot (r - p) \geq 0 \\ p - r, & \text{if } D'(p) \cdot (r - p) < 0 \end{cases} \quad (8)$$

where $L(p, r)$ relates to the line segment between p and r and guarantees that θ_1 is an acute angle. The cost $f_{D_1}(p, r)$ has a small value when r is similar to p , which implies that these two pixels have similar gradient directions. When the current candidate boundary pixel p and neighboring pixel r are both on the boundary, $f_{D_1}(p, r)$ should have a small value, so as to distinguish the similarity of the gradient directions of p and r sufficiently. But, when the boundary is blurred or noisy, more information is needed. Thus, we suggest a new gradient direction cost, $f_{D_2}(p, r)$, for detecting smaller gradient-direction differences. Figure 4 shows that when $l(p, r)$ includes the new cost component $f_{D_2}(p, r)$, 2D live wire gives a better result.

FIGURES 3 and 4 here.

2.2 Proposed 3D Live-Wire Method

We now propose a new 3D live-wire method that enables rapid, flexible definition of 3D ROIs. The proposed method only requires the human operator to interact with a few 2D image sections in general, and it can be used successfully by a novice after a short training period. The basic idea is as follows. First, 2D live wire is run on a few sections oriented in one or two of the standard

orthogonal viewing directions (transverse, sagittal, and coronal) [3] — call these directions D_{or1} and D_{or2} . This establishes a few seeds on all 2D sections in the remaining orthogonal direction D_{lw} . Next, an automatic 3D live-wire process derives the 2D boundaries constituting the 3D ROI on all sections in direction D_{lw} .

Assume that the desired 3D ROI S consists of a discrete set of boundary points $S(x, y, z)$. This 3D collection of boundary points can also be viewed as a set of 2D boundaries $S_{2D}(i)$ oriented in orthogonal direction D and given by the set

$$S = \{S_{2D}(i_{min}), S_{2D}(i_{min} + 1), \dots, S_{2D}(i), \dots, S_{2D}(i_{max} - 1), S_{2D}(i_{max})\}$$

where i_{min} and i_{max} denote the smallest and largest section indices that S resides on for direction D . $S_{2D}(i) = S(x, y, i)$ for the transverse direction, $S_{2D}(i) = S(x, i, z)$ for coronal, and $S_{2D}(i) = S(i, y, z)$ for sagittal, and the direction D corresponds to any of the designated directions D_{or1} , D_{or2} , or D_{lw} . The 3D live-wire method appears below, with Figure 5 schematically illustrating the basic steps and Figure 6 giving the automatic 3D live-wire process:

1. Study the shape of ROI S by scrolling through the 2D sections of the given 3D image I in the standard orthogonal viewing directions (transverse, sagittal, and coronal). From this observation, determine the orthogonal direction — the 3D live-wire direction D_{lw} — for later running an automated 3D live-wire process (Figure 5a).
2. Apply the improved 2D live-wire method of Section 2.1 to a few representative 2D sections in either one or both of the other orthogonal directions, D_{or1} and D_{or2} . Because $D_{or1} \perp D_{lw}$ and $D_{or2} \perp D_{lw}$, the 2D boundaries from this step provide ROI boundary seeds on all sections in the 3D live-wire direction D_{lw} (Figure 5b-d).
3. Again using 2D live wire, define a reference 2D ROI boundary $S_{2D}(i)$ for one section in direction D_{lw} , $i_{min} \leq i \leq i_{max}$ (Figure 5e). Per the 2D live-wire algorithm, this step includes a 2D rectangular working region that liberally encloses the ROI.
4. Run automatic 3D live wire (Figure 6) in 3D live-wire direction D_{lw} , starting from reference 2D boundary $S_{2D}(i)$ and drawing upon the previously found boundary seeds of step 2. Use the 2D rectangular working region of step 3 for all sections during this process.
5. Study the segmented S and interactively fix incorrectly segmented sections using the 2D live wire. This gives the final segmented ROI S .

FIGURES 5 and 6 here.

As Section 3 demonstrates, flexibility exists in the selection of D_{lw} . Yet, a careful study of the 3D ROI in step 1, before choosing D_{lw} and also before running 2D live wire on a few selected sections in step 2, is important for ensuring a good result. The user should have a reasonable impression of an ROI to understand potential complexities in its shape before beginning any processing.

As illustrated in Figure 5b-c, during step 2, each section processed with the 2D live wire in direction D_{or1} or D_{or2} provides at least two candidate seeds for each section in direction D_{lw} . The number of sections to process in step 2 is subject to the user's judgment. But we have found that a few representative sections spaced evenly over an ROI's extent and typically focused within the ROI's (better-defined) central core to be sufficient in our tests of Sections 3.2-3.3 and in our live bronchoscopy studies alluded to in Section 3.5. An empirical rule of thumb is to attempt to regularly distribute seeds over all ROI sections in direction D_{lw} ; there should not be too many seeds allocated to one part of a boundary, while having too few for other boundary parts.

The reference boundary suggested in step 3 is strictly speaking not necessary, because the seeds provided by the slices analyzed in step 2 provide starting conditions for the automatic 3D live-wire process. But we have found that the reference boundary $S_{2D}(i)$ makes subsequent automatic processing more reliable. In addition, step 3 introduces the important working rectangle that limits the analysis on all sections in direction D_{lw} , reducing computation time.

In step 4, the automatic 3D live-wire process (Figure 6) begins with 2D section $S_{2D}(i + 1)$, adjacent to the given reference boundary $S_{2D}(i)$. Depending on D_{lw} , this section can be oriented either below (D_{lw} =transverse), to the left (D_{lw} =sagittal), or behind (D_{lw} =coronal) $S_{2D}(i)$. At the beginning, the available seeds on this section must be ordered, so that the 2D live-wire step, which is embedded in the 3D process, can define a proper closed 2D boundary. This ordering is done by noting how the known seeds on $S_{2D}(i + 1)$ relate to a projection of the seeds on the reference boundary. After a 2D boundary is completed for section $S_{2D}(i+1)$, this new boundary serves as the reference boundary and processing moves on to next adjacent section $S_{2D}(i + 2)$. Similar steps are then undertaken for sections $S_{2D}(i - 1)$, $S_{2D}(i - 2)$, etc., oriented either above (D_{lw} =transverse), to the right (D_{lw} =sagittal), or in front of (D_{lw} =coronal) the initial given reference boundary $S_{2D}(i)$. Automatic processing stops in both directions, depending on two conditions: (1) The number of known seeds on a section is ≤ 3 , a number that is not suitable for constructing a closed contour;

and an optional criterion (2) whereby the user a priori sets i_{min} and i_{max} — see the discussion below for Figure 8.

Since the 3D ROI can have any possible shape, the sections previously processed in step 2 in directions, D_{or1} and D_{or2} , can produce multiple connected “degenerate” seeds on a section in direction D_{lw} — see Figure 7. To correct these situations, the current reference boundary is projected onto the current section of interest and the distance between seeds and the projected boundary is calculated. This distance, combined with local cost information, is used to remove incorrect and misleading seed candidates. Section 3 points out that the automatic 3D process of step 4 in general runs quickly compared to the other steps of the complete 3D method.

FIGURE 7 here.

2.3 Computer Implementation

To comment on step 5 of the 3D method, we now describe our graphical user interface (GUI) tool containing the proposed live-wire methods (Figure 8). As shown in the figure, the GUI provides considerable flexibility for defining 2D and 3D ROIs:

1. “Live Wire Control” — General control options for both 2D and 3D live-wire operations. Cost parameters are set here, basic region filling can be done, and an undo feature exists to delete poor or accidentally produced results.
2. “3D Live Wire” — The top (i_{min}) and bottom (i_{min}) sections for running automatic 3D live wire can be optionally set. “3D Seg” initiates the automatic 3D live-wire process, while “3D Fill” enables filling of a segmented 3D ROI.
3. “ROI Operations” — These general functions supplement the semi-automatic processing of the live wire by enabling manual slice tracing, region erase/copy/paste operations, and a general undo feature.

The tool can be run on transverse, sagittal, and coronal sections and constitutes part of a system we have been devising for MDCT-based planning and guidance of bronchoscopy [37, 38] (see Section 3.5). Per step 5 of the basic 3D live-wire method, the final ROI boundaries produced by the automatic 3D live-wire process of step 4 need to be examined for errors. Typically, a few slices on the ends of an ROI might exhibit errors. Using the GUI of Figure 8, for example, we can fix

these slices by deleting poor results and redoing them using either the 2D live wire or manual slice tracing. Parts of contours can also be fixed either manually or semi-automatically. Finally, our implementation enables construction of a complex ROI consisting of multiple parts. For example, if an ROI branches from one part into two disconnected parts at section $i = i_{max}$ in direction D_{lw} , then two additional runs of the 3D live-wire method can be run for each part by setting a new i_{min} for these runs equal to the previous i_{max} .

FIGURE 8 here.

3 Results

This section presents a series of tests that illustrate the efficacy and application of the live-wire methods. Section 3.1 first outlines the basic form of the tests and performance factors used for evaluating method performance. Next, Sections 3.2 and 3.3 give 2D and 3D segmentation results, while Section 3.4 addresses computational efficiency. Finally, Section 3.5 illustrates an application of 3D live wire for planning and guiding bronchoscopic biopsy, an important procedure for human lung-cancer assessment.

3.1 Segmentation Test Overview and Performance Factors

Five human operators $\mathcal{O} \in \{\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3, \mathcal{O}_4, \mathcal{O}_5\}$ were given the task of segmenting a preselected set of ROIs $R \in \{R_1, R_2, R_3, R_4, R_5, R_6\}$ depicted in typical 3D MDCT pulmonary images. As discussed more in Sections 3.2 and 3.3, both 2D and 3D ROIs, which varied in shape and reflected the different levels of complexity found in medical image-segmentation tasks, were selected for the tests. For both the 2D and 3D tests, each operator \mathcal{O} repeated a segmentation method over three separate trials $\mathcal{E} \in \{\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3\}$.

When segmenting a particular ROI, an operator was free to decide where to begin a segmentation. In addition, for the 3D ROIs, the operator was free to choose which orthogonal and reference sections to consider. All experiments were run on a Dell Precision 650 workstation, using dual Intel Xeon 3.2GHz processors, 3GB RAM, and Windows XP.

As it is impossible to compare results from many different cost weights, we used the following values for all tests: $w_G = 0.4$, $w_Z = 0.17$, $w_{D_1} = 0.33$, $w_{D_2} = 0.1$, and $T_{lap} = 0.05$. We have found these values to work well for our MDCT chest images both in the tests described below and

in our guided bronchoscopy studies. We have noticed considerable robustness and flexibility in the choice of these weights, without appreciable change in performance. Several points, however, bear comment. All cost components (2-5) that make up $l(p, r)$ in (1) are computed such that they are normalized to range from 0 to 1. In addition, weights were assigned so that they sum to 1. These constraints bound the cost to the range $0 \leq l(p, r) \leq 1$. In principle, the gradient-magnitude cost $f_G(r)$ in (2) is given the greatest weight, as the edge strength tends to be the most telling criterion. The Laplacian zero-crossing cost $f_Z(r)$ of (3) is given a relatively lower weight, because it is binary valued and more susceptible to noise; a larger T_{lap} can conceivably lower this criterion's noise susceptibility.

To measure the efficacy of the methods, we adapted four previously suggested performance factors: intraobserver and interobserver reproducibility, accuracy, and efficiency [19, 21].

Reproducibility measures for a particular ROI R_i the similarity of segmentation results obtained on different trials. Both intraobserver reproducibility (results for the same operator over different trials compared) and interobserver reproducibility (results from different operators compared) were computed. The intraobserver reproducibility of the segmentation for ROI R_i by operator \mathcal{O}_j on trials \mathcal{E}_l and \mathcal{E}_m is given by

$$r_{\text{intra}}(R_i, \mathcal{O}_j, \mathcal{E}_l, \mathcal{E}_m) = 1 - \frac{|B_{R_i \mathcal{O}_j \mathcal{E}_l} \oplus B_{R_i \mathcal{O}_j \mathcal{E}_m}|}{|B_{R_i \mathcal{O}_j \mathcal{E}_l}| + |B_{R_i \mathcal{O}_j \mathcal{E}_m}|}, \quad l \neq m \quad (9)$$

while interobserver reproducibility of the segmentation for R_i between operator \mathcal{O}_j on trial \mathcal{E}_l and operator \mathcal{O}_k on trial \mathcal{E}_m is given by

$$r_{\text{inter}}(R_i, \mathcal{O}_j, \mathcal{O}_k, \mathcal{E}_l, \mathcal{E}_m) = 1 - \frac{|B_{R_i \mathcal{O}_j \mathcal{E}_l} \oplus B_{R_i \mathcal{O}_k \mathcal{E}_m}|}{|B_{R_i \mathcal{O}_j \mathcal{E}_l}| + |B_{R_i \mathcal{O}_k \mathcal{E}_m}|}, \quad j \neq k \quad (10)$$

where $B_{R_i \mathcal{O}_j \mathcal{E}_l}$, represented as a binary mask, is the segmentation result for ROI R_i by operator \mathcal{O}_j on trial \mathcal{E}_l , \oplus is the exclusive-or operator, and $|B|$ represents the sum of the voxels constituting B .

Accuracy measures how the segmentation of a given ROI corresponds to a ground-truth result. The accuracy of a segmentation of ROI R_i by operator \mathcal{O}_j on trial \mathcal{E}_l is calculated as follows:

$$a(R_i, \mathcal{O}_j, \mathcal{E}_l) = 1 - \frac{|B_{R_i \mathcal{O}_j \mathcal{E}_l} \oplus G_i|}{|B_{R_i \mathcal{O}_j \mathcal{E}_l}| + |G_i|} \quad (11)$$

where G_i is a predefined ground truth segmentation of R_i . Since all image data used in this paper arise from real 3D human MDCT chest scans, we had to predefine ground-truth segmentations.

For each ROI R_i , a human expert used the improved 2D live-wire method carefully to produce a corresponding ground-truth segmentation G_i . For the 3D ROIs, an expert produced a ground-truth result by again applying the improved 2D live-wire method slice by slice in a convenient viewing direction; the 2D method enables the greatest control over the produced ground truth and permits the expert to carefully define boundaries at all locations. The expert, who defined the ground truth, did not serve as an operator in the test results reported below and did not assist the other human operators who took part in the tests.

The final performance factor, efficiency, is defined simply as the processing time in seconds required to segment an ROI. For the tests, the human operator was permitted to study a given ROI, be it 2D or 3D, before beginning the segmentation process. This observation time was not included as part of the measured processing time.

3.2 2D Segmentation Results

Three 2D ROIs $R \in \{R_1, R_2, R_3\}$ were selected for 2D segmentation tests, as shown in Table 1 and Figure 9. R_1 , situated above the right upper lung lobe, is a large ROI, with vague boundaries to the right and bottom left. R_2 is a subaortic lymph node exhibiting a blurry bottom-left boundary. Finally, R_3 is a sagittal section of the pulmonary artery near the heart. Its boundary close to the aorta is difficult to define.

	Section Orientation and Number	MDCT Image Information			
		Image Dimensions	Δx	Δy	Δz
R_1	transverse, $z = 9$	$512 \times 512 \times 475$	0.52	0.52	0.60
R_2	transverse, $z = 163$	$512 \times 512 \times 481$	0.64	0.64	0.60
R_3	sagittal, $x = 275$	$512 \times 512 \times 597$	0.67	0.67	0.50

Table 1: MDCT images used for 2D ROIs, R_1 , R_2 , and R_3 . Sampling intervals, Δx , Δy , and Δz , have units mm. The images for R_1 (p2h044) and R_2 (p2h031) were generated by a Philips 4-detector MDCT scanner, while the image for R_3 (bb003) was produced with contrast enhancement by a Siemens Sensation 16-detector MDCT scanner.

FIGURE 9 here.

Tables 2-3 give performance measures, while Figure 10 shows sample segmentation results. All operators exceeded nearly 98% intraobserver reproducibility over their repeated trials with virtually no variance between trials (Table 2 and Table 3, row 1). In addition, for interobserver variability, each operator gave virtually indistinguishable performance when compared to other

operators (Table 3, row 2). Further, the results produced by the operators over multiple trials showed very strong agreement with ground truth segmentations (Table 3, row 3). In fact, the minimal variations and high accuracy values shown in Table 3 attest to the remarkable consistency of the method. These results carried over for all three 2D ROIs considered. Thus, the improved 2D live-wire method enables an operator to generate highly reproducible results, shows essentially no dependence on the operator, and is very accurate when compared to ground truth. (Reference [41] provides further confirming results.)

	R_1		R_2		R_3	
	μ	σ	μ	σ	μ	σ
\mathcal{O}_1	98.0	0.29	99.3	0.26	98.0	0.30
\mathcal{O}_2	98.7	0.38	99.4	0.13	98.9	0.40
\mathcal{O}_3	97.7	0.63	98.6	0.34	98.6	0.48
\mathcal{O}_4	97.9	0.27	97.3	0.37	98.2	0.12
\mathcal{O}_5	98.0	0.39	98.7	0.27	97.9	0.50

Table 2: 2D segmentation intraobserver reproducibility $r_{\text{intra}}(R_i, \mathcal{O}_j, \mathcal{E}_l, \mathcal{E}_m)$. The operators used the improved 2D live wire to segment the 2D ROIs over three independent trials. A value μ gives an average of $r_{\text{intra}}(R_i, \mathcal{O}_j, \cdot, \cdot)$ expressed as a percentage of pixels in agreement over all three trials, with σ being the corresponding standard deviation.

	R_1		R_2		R_3	
	μ	σ	μ	σ	μ	σ
$r_{\text{intra}}(R_i)$	98.0	0.52	98.7	0.80	98.3	0.52
$r_{\text{inter}}(R_i)$	97.9	0.75	98.5	0.72	98.0	0.60
$a(R_i)$	98.4	0.55	98.8	0.56	97.3	0.72

Table 3: 2D segmentation performance measures for the improved 2D live-wire method: intraobserver reproducibility $r_{\text{intra}}(R_i)$, interobserver reproducibility $r_{\text{inter}}(R_i)$, and accuracy $a(R_i)$. The average μ and standard deviation σ values for $r_{\text{intra}}(R_i)$ combine the results of all operators from Table 2. To arrive at statistics for $r_{\text{inter}}(R_i)$, we first computed $r_{\text{inter}}(R_i, \mathcal{O}_j, \mathcal{O}_k, \mathcal{E}_l, \mathcal{E}_m)$ for each operator \mathcal{O}_j . This was done by comparing the segmentation results of operator \mathcal{O}_j to all other operators $\mathcal{O}_k, k = 1, \dots, 5, k \neq j$, over all segmentation trials. We next combined the individual operator values to arrive at overall statistics, μ and σ , for interobserver reproducibility. μ represents the average percentage of segmented pixels in agreement over all trials, with σ corresponding to the standard deviation. Finally, $a(R_i)$ combines all separate accuracy measures $a(R_i, \mathcal{O}_j, \mathcal{E}_l)$ for all five human operators \mathcal{O}_j over three independent trials \mathcal{E}_l . μ represents the average percentage of segmented pixels in agreement with ground-truth segmentation results over all operators and trials, with σ being the corresponding standard deviation.

FIGURE 10 here.

3.3 3D Segmentation Results

Three 3D ROIs $R \in \{R_4, R_5, R_6\}$ were selected for 3D segmentation tests, as summarized in Table 4 and Figure 11. ROI R_4 is a small suspect cancer nodule, having a clear object boundary in all viewing directions. R_5 , also a suspect cancer nodule, is bigger and more complicated than R_4 . Most of its transverse slices exhibit a clear boundary, but many of its boundaries on coronal and sagittal sections appear blurry. Finally, R_6 represents a major part of the large aorta.

	Section Ranges			MDCT Image Information			
	Transverse (z)	Coronal (y)	Sagittal (x)	Image Dimensions	Δx	Δy	Δz
R_4	418-435	316-331	361-374	$512 \times 512 \times 597$	0.74	0.74	0.60
R_5	215-260	308-355	363-413	$512 \times 512 \times 479$	0.586	0.586	0.60
R_6	170-299	204-348	218-312	$512 \times 512 \times 597$	0.67	0.67	0.50

Table 4: MDCT images used for the 3D ROIs R_4 , R_5 , and R_6 . A “section range” denotes the range of sections in a given orthogonal direction where the ROI exists. Sampling intervals, Δx , Δy , and Δz , have units mm. The images for R_4 (p1h019) and R_5 (p1h005) were generated by a Philips 4-detector MDCT scanner, while the image for R_6 is the same data set used for R_3 (bb003).

FIGURE 11 here.

We again tested the same five human operators as in Section 3.2. Each operator ran three trials in each of the orthogonal viewing directions — transverse, sagittal, and coronal — for D_{lw} using the 3D live wire approach for R_4 and R_5 . Thus, each operator performed nine segmentation trials total for R_4 and R_5 . For R_4 , only one section from each orthogonal direction, D_{or1} and D_{or2} , was processed using 2D live wire to produce pre-defined seeds for the automatic 3D live-wire process. For R_5 , two orthogonal sections in each direction, D_{or1} and D_{or2} , were processed using 2D live wire before running automatic 3D live wire. Because the aorta is a long narrow region that extends vertically within the chest over a long range of slices, R_6 can only be segmented practically with $D_{lw} = \text{transverse}$ via the 3D live-wire method. For R_6 , an operator first ran 2D live wire on four to five coronal and sagittal sections before running automatic 3D live wire in the transverse direction.

Tables 5-6 give performance measures, while Figures 12-13 depict sample 2D and 3D segmentation results. The 3D performance nearly matches the 2D performance. All operators exceeded 97% intraobserver reproducibility over their repeated trials with little variance between trials (Table 5 and Table 6, row 1). For interobserver variability, each operator again gave nearly identical performance to other operators (Table 6, row 2). Also, the operator results over multiple trials indicate

strong agreement with ground truth (Table 6, row 3). Therefore, we can arrive at essentially the same conclusions for the 3D live-wire method as we did earlier for 2D live wire: the method enables the production of highly reproducible results, exhibits virtually no operator dependence, and is highly accurate. (Reference [41] again provides additional numerical results.)

	R_4						R_5						R_6	
	Transverse		Coronal		Sagittal		Transverse		Coronal		Sagittal		Transverse	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
\mathcal{O}_1	98.6	0.59	98.9	0.36	97.0	0.71	99.5	0.30	98.4	0.30	99.2	0.25	98.1	0.39
\mathcal{O}_2	98.7	0.58	98.6	0.61	98.0	0.45	99.1	0.70	98.5	0.30	99.1	0.26	98.3	0.45
\mathcal{O}_3	98.1	0.59	98.9	0.44	97.6	0.40	99.6	0.26	98.5	0.52	98.9	0.54	98.0	0.35
\mathcal{O}_4	98.3	0.32	98.5	0.41	97.9	0.23	99.6	0.22	98.0	0.39	97.5	0.98	98.1	0.67
\mathcal{O}_5	97.4	0.43	99.0	0.23	96.8	0.33	99.2	0.36	98.2	0.14	98.4	0.45	97.9	0.83

Table 5: 3D segmentation intraobserver reproducibility $r_{\text{intra}}(R_i, \mathcal{O}_j, \mathcal{E}_l, \mathcal{E}_m)$. For R_4 and R_5 , three independent trials were run of the 3D live-wire algorithm in each of the orthogonal directions (i.e., D_{lw} = transverse, coronal, or sagittal). For R_6 , only D_{lw} = transverse was used. A value μ gives an average of $r_{\text{intra}}(R_i, \mathcal{O}_j, \cdot, \cdot)$ expressed as a percentage of pixels in agreement over all three trials, with σ being the associated standard deviation.

	R_4		R_5		R_6	
	μ	σ	μ	σ	μ	σ
$r_{\text{intra}}(R_i)$	98.1	0.77	98.8	0.73	98.1	0.51
$r_{\text{inter}}(R_i)$	98.0	0.98	98.4	0.76	97.6	0.53
$a(R_i)$	98.1	0.35	98.1	0.75	97.2	0.21

Table 6: 3D segmentation performance measures for the new 3D live-wire method: intraobserver reproducibility $r_{\text{intra}}(R_i)$, interobserver reproducibility $r_{\text{inter}}(R_i)$, and accuracy $a(R_i)$. These aggregate values were computed in the same way as done for the 2D results of Table 3, with μ and σ again denoting overall average and standard deviation. ($r_{\text{intra}}(R_i)$ combines the Table 5 results.)

FIGURES 12-13 here.

3.4 Efficiency of the Live-Wire Methods

This section presents efficiency results for both the 2D and 3D tests described in Sections 3.2-3.3.

Tables 7-8 give measurements for efficiency and also importantly compare the computational efficiency of the improved 2D live-wire method to standard manual slice tracing. Table 7 clearly shows the efficiency of 2D live wire — all ROIs were segmented in under 30 sec on average overall.

Table 8 dramatically demonstrates the considerable efficiency of 2D live wire over manual slice

tracing. Over all ROIs, 2D live wire required a factor of 14 less interaction time than manual slice tracing.

	R_1		R_2		R_3	
	μ	σ	μ	σ	μ	σ
\mathcal{O}_1	35	14.0	11	0	26	7.1
\mathcal{O}_2	20	4.4	13	2.0	16	5.5
\mathcal{O}_3	22	1.0	11	1.0	21	2.1
\mathcal{O}_4	22	6.0	19	7.6	17	2.3
\mathcal{O}_5	18	4.0	11	1.5	13	1.2
Overall	23.4	8.8	13	4.5	18.6	5.9

Table 7: 2D live-wire segmentation time (efficiency) in seconds; μ and σ are the mean and standard deviation of segmentation time over three trials.

	R_1	R_2	R_3
2D Live Wire	23.4	13	18.6
Manual Slice Tracing	310	192	278

Table 8: Average 2D segmentation times in seconds for three 2D ROIs using two methods for segmentation: the improved 2D live-wire method and manual slice tracing. For manual slice tracing, we required an accuracy rate $>95\%$ to consider its results in the efficiency comparison.

Table 9 gives efficiency results for the 3D live-wire method. With the 3D live wire, R_4 , the smallest ROI, proved to be the quickest to segment, with R_5 taking the second longest time, and R_6 , the complex aorta, requiring the most time. For R_4 and R_5 , no particular live-wire direction D_{lw} gave any appreciable advantage, probably because of the relative 3D symmetry of these ROIs. An operator required on the average of 10 sec of interaction time for each 2D orthogonal (D_{or1}, D_{or2}) section of R_4 , an ROI situated on 18 sections, and 15 sec of interaction time for each 2D orthogonal section of R_5 , a slightly larger 45-section ROI. The automatic 3D live-wire process required approximately 1 sec for R_4 and < 10 sec for R_5 . Thus, the interaction time dominated the overall computation time, without an undo delay from the 3D automated processing. (We did not attempt to optimize our computer code for the automated processing.)

R_4 's 3D live-wire segmentations had on the order of 1-2 incorrect 2D boundaries, which were redefined with the 2D live wire; for all trials, it took roughly 30 seconds to fix these incorrect boundaries. R_5 's 3D live-wire segmentation had on the order of 1-5 incorrect boundaries, with 30 seconds to 2 minutes of 2D live-wire post-processing required on average to correct. For R_6 , 2-10 sections of the 3D segmentation typically required post processing, costing less than 4 minutes on

average to define the corresponding proper boundaries using 2D live wire. It is important to note that these incorrect boundaries were always on the ends of an ROI and also were considerably tapered (almost to the point of having little spatial support) from the main ROI body; i.e., these cross-sections were regions of large local change (the ROI is terminating!). Hence, these boundaries were relatively insignificant. Because postprocessing required discretion and because the operator always has to examine the extracted ROI regardless of whether there were erroneous sections or not, we did not include this time in 3D live-wire times.

	R_4						R_5						R_6	
	Transverse		Coronal		Sagittal		Transverse		Coronal		Sagittal		Transverse	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
\mathcal{O}_1	41	4.9	36	5.3	33	3.6	91	12.1	100	14.5	101	9.1	561	21.9
\mathcal{O}_2	28	2.1	30	2.1	31	4.6	86	7.5	86	1.7	79	7.8	442	38.1
\mathcal{O}_3	34	7.9	34	7.0	27	3.6	116	23.2	88	9.9	93	13.5	468	16.3
\mathcal{O}_4	37	10.2	36	6.5	31	4.9	104	13.7	97	7.8	98	3.1	496	26.9
\mathcal{O}_5	36	7.5	31	4.0	34	8.7	93	3.1	94	14.8	101	10.3	477	61.3
Overall	35.5	7.5	33.3	5.1	31.3	5.2	98	16.0	92.9	10.8	94.5	11.70	489	51.8

Table 9: 3D live-wire segmentation time (efficiency) in seconds; μ and σ are the mean and standard deviation of segmentation time over all trials. All times exclude 2D live-wire post-processing.

Table 10 compares the segmentation efficiency of three separate methods for 3D segmentation: 3D live wire; 2D live wire, whereby a 3D ROI was segmented section by section in a particular orthogonal direction; and 2D manual slice tracing. Over all cases, the 2D live-wire method is at least a factor 4 faster than manual slice tracing, while 3D live wire is at least 7.2 times faster than 2D live wire and 28 times faster than manual slice tracing, including post-processing time.

	R_4			R_5			R_6
	Transverse	Coronal	Sagittal	Transverse	Coronal	Sagittal	Transverse
3DLW	36s	33s	31s	98s	93s	94s	8m9s
2DLW	5m11s	4m7s	3m42s	16m36s	19m15s	13m14s	1h20m24s
MST	20m9s	17m6s	14m36s	1h31m22s	1h48m35s	1h42m33s	5h29m20s

Table 10: Average 3D segmentation times in hours (h), minutes (m), and seconds (s) using three different segmentation methods: 3D live wire (3DLW), 2D live wire (2DLW), and manual slice tracing (MST). For manual slice tracing, we required an accuracy rate $>90\%$ to consider its results in the efficiency comparison.

For both the 2D and 3D results, only the expert operator, who produced the ground truth,

performed manual slice tracing. We required the expert to give accuracy results with respect to the ground truth segmentations G_i of at least 95% for the 2D ROIs and 90% for the 3D ROIs. The expert needed to perform two trials for a few of the ROIs to achieve the required accuracy. It is, of course, possible to perform manual slice tracing “quickly,” but hurried results are generally unsatisfactory and not usable in practice. The modest accuracy requirement we imposed on the manual slice-tracing results forced a disciplined cooperative effort from the expert.

The efficiency discrepancy between the 2D results and 3D results warrants comment. As indicated above, purely 2D live-wire analysis required a factor of 14 less interaction for the 2D ROIs and a factor of only 4 less interaction time for the 3D ROIs. This occurred for a few reasons. First, switching from one section to the next for the 3D ROIs imposes a small delay. In addition, much more importantly, the operator tends to get fatigued when confronted with the task of using 2D live wire on a large number of sections. For the larger ROIs, R_5 and R_6 , the operators experienced fatigue and their hands became tired from moving the mouse. Thus, as expected, the 3D live-wire method is far preferred for 3D ROIs over the 2D live wire.

3.5 Application to the Planning and Guidance of Bronchoscopy

Bronchoscopic biopsy of the mediastinal lymph nodes is an important procedure for staging lung cancer [2, 7]. This procedure requires prior off-line 3D MDCT-based preplanning followed by live bronchoscopic guidance. For the past several years, we have been devising a system for MDCT-based planning and guidance of bronchoscopy [37, 38]. A substantial aspect of this problem is the definition of the 3D central-chest lymph nodes. Unfortunately, the lymph nodes can have virtually any shape and size and widely varying intensity characteristics. Thus, no proven robust automated technique exists for defining these structures. Therefore, like others [7], we had to resort to tedious manual slice tracing in early tests to define these 3D structures [37].

In our current efforts, we employ the 3D live-wire method to rapidly define the central chest lymph nodes — see Figure 14. With the 3D live wire, the physician is able to define these complex structures far more efficiently — in only a few minutes — than in previous efforts, making the off-line MDCT-based preplanning stage much less demanding on the physician’s time. Notice how Figure 14a confirms the greatly varying shape, size, and location of the lymph nodes. After preplanning is complete, we then employed our system to perform MDCT-based guidance of the

live bronchoscopy. For each preplanned ROI, the physician would follow the prescribed centerline until the ROI's location is reached. As shown in Figure 14b, we employ a fusion of the predefined 3D MDCT data onto the registered bronchoscopic video to provide unambiguous live guidance information. Note that the lymph nodes are situated outside the airways and not visible in the video. But by registering and fusing the MDCT and video sources, our system can provide a view of these extraluminal ROIs on the live video. Refer to [38] for further discussion.

FIGURE 14 here.

4 Conclusion

Fully automated image segmentation still remains unsolved for most 3D medical image processing applications and certainly is not reliable for real mission-critical routine clinical use. Since even the most robust automated image-segmentation methods tend to fail when they are applied to other region or image types, a fail-safe image-segmentation technique is essential for successfully performing tasks in a real clinical scenario, where failure is unacceptable. Previously, investigators were forced to rely on the unsatisfactory manual slice tracing. But the live wire greatly improves upon this approach in terms of interaction time, reliability, and reproducibility. It also has the vital property that it is “guaranteed to work.” This also makes the live wire arguably the best available method currently for producing ground truth image-segmentation results for testing new automated methods.

The experimental results clearly show that our proposed live-wire methods enable efficient definition of 2D and 3D ROIs in 3D MDCT images. They also demonstrate that the generated segmentations are robust and are virtually independent of the human operator producing them. Our methods, motivated by the seminal early live-wire work of Mortensen, Barrett, and Udupa, have also been incorporated into a convenient graphical user interface tool for flexible definition of regions.

The proposed 2D method incorporates an improved cost to give more robust results than previous proposals, and it also incorporates a working region to reduce the computation of the graph search. The 3D method offers considerable flexibility in that the operator need only interact with a few sections in general, with most of the computation performed automatically. Some mild fixing of errant slices is required in general, but these only occur on the ends of an ROI, where the ROI

is nearly tapered to zero support.

The interactive segmentation tool, which incorporates the 2D and 3D live-wire algorithms, is easily operated by a novice after a short training period. As the methods are intuitive and visual, a new operator was always able to quickly define ROIs. We point out that the 3D method has no restriction on the orthogonal sections, reference section, and bounding area selected (but we do assume that an operator “cooperates” and studies the image data reasonably).

We focused our efforts on 3D MDCT chest images. This is important, as effective 3D image segmentation is vital for defining complex irregular diagnostic ROIs for radiologic image-based disease assessment, treatment planning, and follow-on surgical/bronchoscopic guidance. But, as our methods have no modality or domain dependence, we believe the methods are generally applicable. As ROI definition in general is vital in most areas of 3D medical image analysis — not just the chest — the methods could have wide applicability.

5 Acknowledgments

This work was partially supported by grants #CA074325 and #CA091534 from the NIH National Cancer Institute.

References

- [1] E. A. Kazerooni, “High resolution CT of the lungs,” *Am. J. Roentgenology*, vol. 177, no. 3, pp. 501–519, Sept. 2001.
- [2] A. D. Sihoe and A. P. Yim, “Lung cancer staging,” *J. Surgical Research*, vol. 117, no. 1, pp. 92–106, Mar. 2004.
- [3] N. C. Dalrymple, S. R. Prasad, M. W. Freckleton, and K. N. Chintapalli, “Introduction to the language of three-dimensional imaging with multidetector CT,” *Radiographics*, vol. 25, no. 5, pp. 1409–1428, Sept.-Oct. 2005.
- [4] A. P. Kiraly, E. A. Hoffman, G. McLennan, W. E. Higgins, and J. M. Reinhardt, “3D human airway segmentation methods for clinical virtual bronchoscopy,” *Academic Radiology*, vol. 9, no. 10, pp. 1153–1168, Oct. 2002.
- [5] C. I. Fetita, F. Prêteux, C. Beigelman-Aubry, and P. Grenier, “Pulmonary airways: 3-D reconstruction from multislice CT and clinical investigation,” *IEEE Trans. on Medical Imaging*, vol. 23, no. 11, pp. 1353–1364, Nov. 2004.
- [6] M. Brown, M. McNitt-Gray, J Goldin, R Suh, J Sayre, and D Aberle, “Patient-specific models for lung nodule detection and surveillance in CT images,” *IEEE Trans. Med. Imaging*, vol. 20, no. 12, pp. 1242–1250, Dec. 2001.
- [7] H. P. McAdams, P. C. Goodman, and P. Kussin, “Virtual bronchoscopy for directing transbronchial needle aspiration of hilar and mediastinal lymph nodes: a pilot study,” *Am. J. Roentgenology*, vol. 170, pp. 1361–1364, May 1998.

- [8] A. P. Kiraly, J. P. Helferty, E. A. Hoffman, G. McLennan, and W. E. Higgins, “3D path planning for virtual bronchoscopy,” *IEEE Trans. Medical Imaging*, vol. 23, no. 9, pp. 1365–1379, Sept. 2004.
- [9] W.E. Higgins, N. Chung, and E.L. Ritman, “LV-chamber extraction from 3-D CT images: accuracy and precision,” *Comput. Med. Imaging Graph.*, vol. 16, no. 1, pp. 17–26, Jan.-Feb. 1992.
- [10] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *Int. J. Computer Vision*, vol. 1, no. 4, pp. 321–331, Jan. 1988.
- [11] L. D. Cohen and R. Kimmel, “Global minimum for active contour models: A minimal path approach,” *Int. J. Computer Vision*, vol. 24, no. 1, pp. 57–78, Aug. 1997.
- [12] T. Kunert, T. Heimann, A. Schröter, M. Schöbinger, T. Böttger, M. Thorn, I. Wolf, U. Engelmann, and H. P. Meinzer, “An interactive system for volume segmentation in computer-assisted surgery,” in *SPIE Medical Imaging 2004: Visualization, Image-Guided Procedures, and Display*, R. L. Galloway, Ed., May 2004, vol. 5367, pp. 799–809.
- [13] T. McInerney and M. R. Akhavan-Sharif, “Sketch initialized snakes for rapid, accurate and repeatable interactive medical image segmentation,” in *3rd IEEE International Symposium on Biomedical Imaging 2006: Macro to Nano*, Apr. 2006, pp. 398–401.
- [14] E. N. Mortensen, B. S. Morse, W. A. Barrett, and J. K. Udupa, “Adaptive boundary detection using “live-wire” two-dimensional dynamic programming,” *IEEE Proc. of Computers in Cardiology*, vol. 11, no. 14, pp. 635–638, Oct. 1992.
- [15] J. K. Udupa, S. Samarasekera, and W. A. Barrett, “Boundary detection via dynamic programming,” in *SPIE Visualization in Biomedical Computing '92*, R. A. Robb, Ed., Sept. 1992, vol. 1808, pp. 33–39.
- [16] E. N. Mortensen and W. A. Barrett, “Intelligent scissors for image composition,” in *Proc. ACM SIGGRAPH95: 22nd Int. Conf. Computer Graphics and Interactive Techniques*, Aug. 1995, pp. 191–198.
- [17] A. X. Falcão, J. K. Udupa, S. Samarasekera, and B. E. Hirsch, “User-steered image boundary segmentation,” in *SPIE Medical Imaging 1996: Image Processing*, M. H. Loew and K. M. Hanson, Eds., Apr. 1996, vol. 2710, pp. 278–288.
- [18] A. X. Falcão and J. K. Udupa, “Segmentation of 3D objects using live-wire,” in *SPIE Medical Imaging 1997: Image Processing*, K. M. Hanson, Ed., Apr. 1997, vol. 3034, pp. 228–239.
- [19] A. X. Falcão, J. K. Udupa, S. Samarasekera, and S. Sharma, “User-steered image segmentation paradigms: Live wire and live lane,” *Graphical Models and Image Processing*, vol. 60, no. 4, pp. 233–260, July 1998.
- [20] E. N. Mortensen and W. A. Barrett, “Interactive segmentation with intelligent scissors,” *Graphical Models and Image Processing*, vol. 60, no. 5, pp. 349–384, Sept. 1998.
- [21] A. X. Falcão and J. K. Udupa, “A 3D generalization of user-steered live-wire segmentation,” *Medical Image Analysis*, vol. 4, no. 4, pp. 389–402, Dec. 2000.
- [22] A. X. Falcão, J. K. Udupa, and F. K. Miyazawa, “An ultra-fast user-steered live-wire segmentation paradigm: Live wire on the fly,” *IEEE Trans. on Medical Imaging*, vol. 19, no. 1, pp. 55–62, Jan. 2000.
- [23] E. N. Mortensen, L. J. Reese, and W. A. Barrett, “Intelligent selection tools,” *IEEE Proc. Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 776–777, June 2000.
- [24] W. A. Barrett, L. J. Reese, and E. N. Mortensen, “Intelligent segmentation tools,” *IEEE Proc. 2002 International Symposium on Biomedical Imaging*, pp. 217–220, July 2002.
- [25] T. Haenselmann and W. Effelsberg, “Wavelet-based semi-automatic live-wire segmentation,” in *SPIE Electronic Imaging 2003: Human Vision and Electronic Imaging VIII*, Bernice E. Rogowitz and Thrasyvoulos N. Pappas, Eds., June 2003, vol. 5007, pp. 260–269.
- [26] H.-W. Kang and S.-Y. Shin, “Enhanced lane: Interactive image segmentation by incremental path map construction,” *Graphical Models*, vol. 64, no. 5, pp. 282–303, Sept. 2003.

- [27] A. Chodorowski, U. Mattsson, M. Langille, and G. Hamarneh, "Color lesion boundary detection using live wire," in *SPIE Medical Imaging 2005: Image Processing*, J. M. Fitzpatrick and J. M. Reinhardt, Eds., Apr. 2005, vol. 5747, pp. 1589–1596.
- [28] W. Wiclawek, "Live-wire method with FCM classification," *IEEE Proc. International Conference on Mixed Design of Integrated Circuits and System*, pp. 756–76, June 2006.
- [29] W. Wiclawek, "Live-wire method with wavelet cost map definition for MRI images," 2006, In IFAC Workshop on Programmable Devices and Embedded Systems, PDeS 2006, Brno.
- [30] C. Rose and K. Binder, *SAMS Teach Yourself Adobe Photoshop CS2 in 24 Hours*, Sams Publishing, Indianapolis, IN, 2005.
- [31] A. Schenk, G. Prause, and H.-O. Peitgen, "Efficient semiautomatic segmentation of 3D objects in medical images," in *MICCAI'00: Proc. of the Third International Conference on Medical Image Computing and Computer-Assisted Intervention*, London, UK, 2000, pp. 186–195, Springer-Verlag.
- [32] S. König and J. Hesser, "3D live-wires on pre-segmented volume data," in *SPIE Medical Imaging 2005: Image Processing*, J. M. Fitzpatrick and J. M. Reinhardt, Eds., Apr. 2005, vol. 5747, pp. 1674–1681.
- [33] S. König and J. Hesser, "3D live-wires on mosaic volumes," *Stud Health Technol Inform.*, vol. 119, pp. 264–266, 2006.
- [34] Z. Salah, J. Orman, and D. Bartz, "Live-wire revisited," 2005, In Workshop Bildverarbeitung für die Medizin, Berlin.
- [35] G. Hamarneh, J. Yang, C. McIntosh, and M. Langille, "3D live-wire-based semi-automatic segmentation of medical images," in *SPIE Medical Imaging 2005: Image Processing*, J. M. Fitzpatrick and J. M. Reinhardt, Eds., Apr. 2005, vol. 5747, pp. 1597–1603.
- [36] A. Souza, J. K. Udupa, G. Grevera, Y. Sun, D. Odhner, N. Suri, and M. D. Schnall, "Iterative live wire and live snake: new user-steered 3D image segmentation paradigms," in *SPIE Medical Imaging 2006: Image Processing*, J. M. Reinhardt and P. W. Pluim, Eds., Mar. 2006, vol. 6144, pp. 61443N1–7.
- [37] J. P. Helferty, A. J. Sherbondy, A. P. Kiraly, and W. E. Higgins, "Computer-based system for the virtual-bronchoscopic guidance of bronchoscopy," *Computer Vision and Image Understanding*, 2007, in press.
- [38] S. A. Merritt, L. Rai, J. D. Gibbs, K.-C. Yu, and W. E. Higgins, "Method for continuous guidance of endoscopy," in *SPIE Medical Imaging 2007: Physiology, Function, and Structure from Medical Images*, A. Manduca and X.P. Hu, Eds., Feb. 2007, vol. 6511.
- [39] T. H. Cormen, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 2001.
- [40] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison Wesley, Reading, MA, 2nd. edition, 2002.
- [41] K. Lu and W. E. Higgins, "Improved 3D live-wire method with application to 3D CT chest images," in *SPIE Medical Imaging 2006: Image Processing*, J. M. Reinhardt and J. P. Pluim, Eds., Mar. 2006, vol. 6144, pp. 189–203.
- [42] K. C. Yu, E. L. Ritman, and W. E. Higgins, "3D model-based vasculature analysis using differential geometry," in *IEEE Int. Symp. on Biomedical Imaging*, Apr. 2004, vol. 1, pp. 177–180.
- [43] K. C. Yu, E. L. Ritman, and W. E. Higgins, "System for 3D visualization and data mining of large vascular trees," in *SPIE Optics East 2005: Three-Dimensional TV, Video, and Display IV*, B. Javadi, F. Okano, and J. Son, Eds., Nov. 2005, vol. 6016, pp. 101–115.

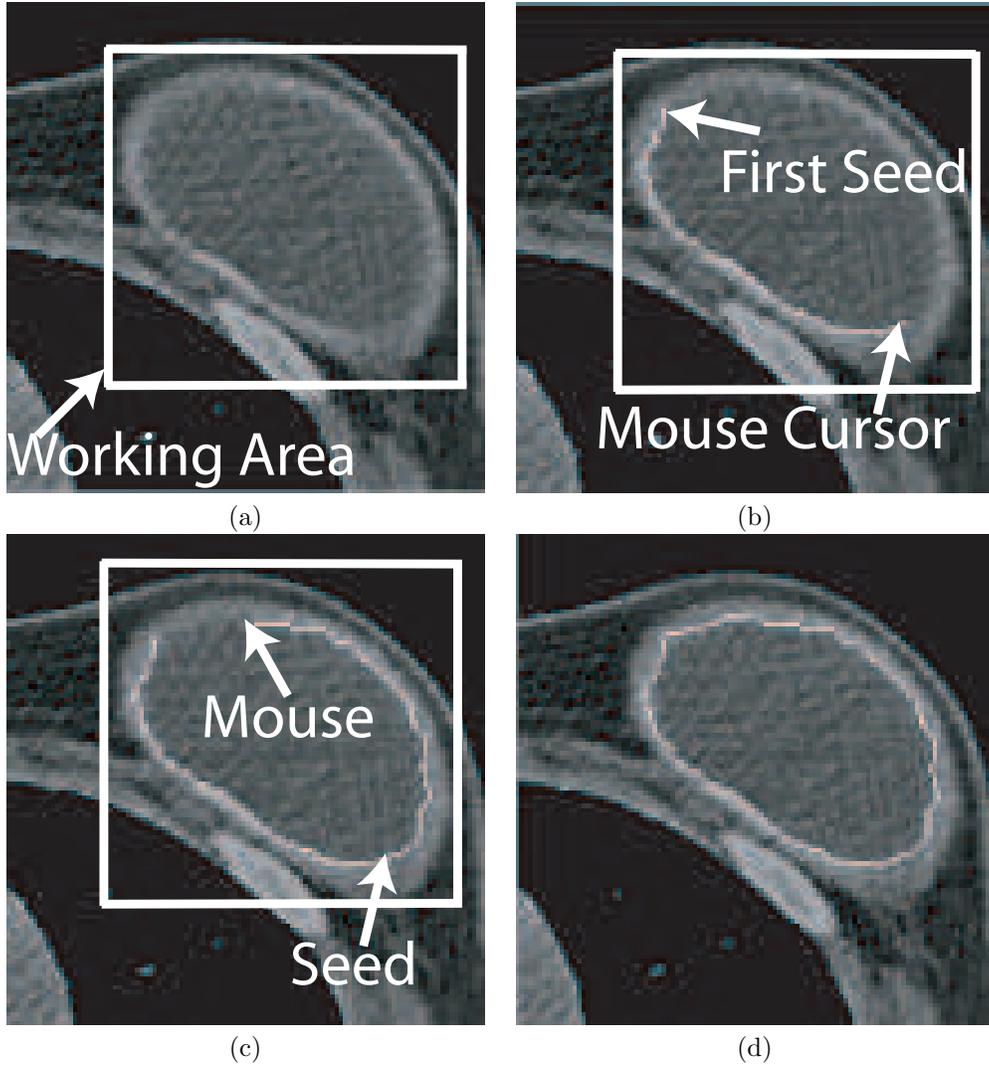


Figure 1: The 2D live-wire processing procedure. (a) The user defines a working rectangular area (white box). (b) The first seed is interactively selected; when the user moves the mouse, the live-wire graph-search algorithm (Figure 2) computes an optimal path in real time from the first seed p to the mouse cursor's location r . (c) The optimal path deposits automatically onto the boundary when a new seed is defined, and a new path between the cursor and the seed is automatically computed. (d) The user performs a double left-mouse button click to complete the closed boundary — the graph search connects the current seed and very first seed. (512×512 2D slice $z = 285$ of human MDCT scan p1h005 slice, generated by a Marconi scanner, with $\Delta x = \Delta y = 0.586\text{mm}$, used in the example.

```

Input:
s                               /* Seed pixel */
l(p, r)                         /* Local cost function for link between pixels p and r */

Data Structures:
L                               /* List of active pixels sorted by total cost (initially empty) */
N(p)                           /* Neighborhood set of pixel p (contains 8-neighbors of p) */
e(p)                           /* Boolean function indicating if p has been processed */
g(p)                           /* Total cost from seed point to pixel p */
pt(p)                          /* Back pointer of pixel p used to indicate the min cost path */

Output
P(p)                           /* Min cost path from pixel p to the seed pixel */

Algorithm:
1. g(s) ← 0; L ← s;           /* Initialize active pixel list with zero-cost seed pixel */
2. While L ≠ ∅ do begin      /* Iterate while active pixel list is not empty */
3.   p ← min(L);             /* Remove minimum-cost pixel from active list and assign it to p */
4.   e(p) ← TRUE;           /* Mark p as processed */
5.   for each r ∈ N(p) such that ¬e(r) do begin
6.     gtemp ← g(p) + l(p, r); /* Compute total cost to a neighbor */
7.     if r ∈ L and gtemp < g(r) then
8.       g(r) ← gtemp; pt(r) ← p; /* Update total cost and set back pointer */
9.     if r ∉ L then begin    /* If neighbor not on list, then assign neighbor's */
10.      g(r) ← gtemp; pt(r) ← p; L ← r; /* total cost, set back pointer, and */
11.    end                    /* place on (or return to) active list */
12.  end
13. end
14. end
15. end
16. Output(P(p))           /* Output min cost path from every pixel p to the seed pixel */

```

Figure 2: Automatic graph-search algorithm for improved 2D live-wire method.

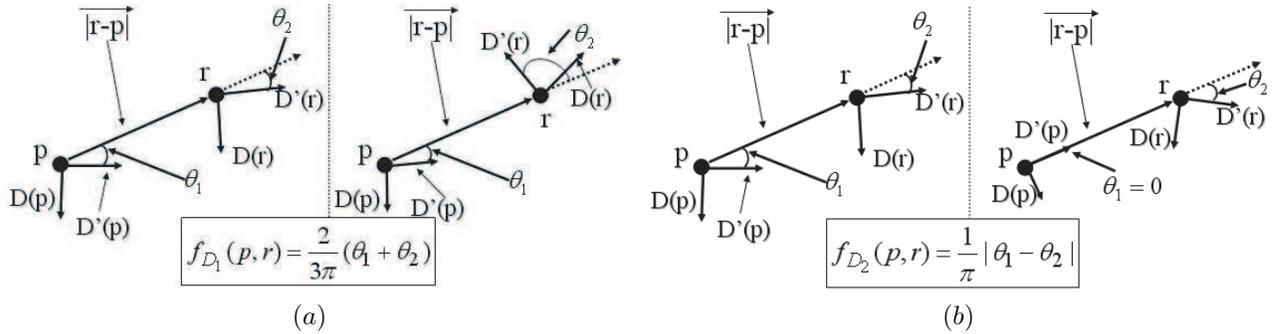


Figure 3: Diagram of gradient direction costs. (a) $D(p)$ and $D(r)$ are the gradient directions of pixels p and r ; $|r-p|$ is the vector from p to r ; $D'(p)$ and $D'(r)$ are vectors orthogonal to $D(p)$ and $D(r)$, and are calculated so that θ_1 is always an acute angle. In this figure, it can be seen that pixels p and r on the left have a more similar gradient direction cost $f_{D_1}(p, r)$ than the corresponding pair of pixels on the right. (b) Pixels p and r on the left have a more similar gradient direction based on secondary gradient-direction cost $f_{D_2}(p, r)$ than the two pixels on the right, even though the $f_{D_1}(p, r)$ values for both pairs are small and similar.

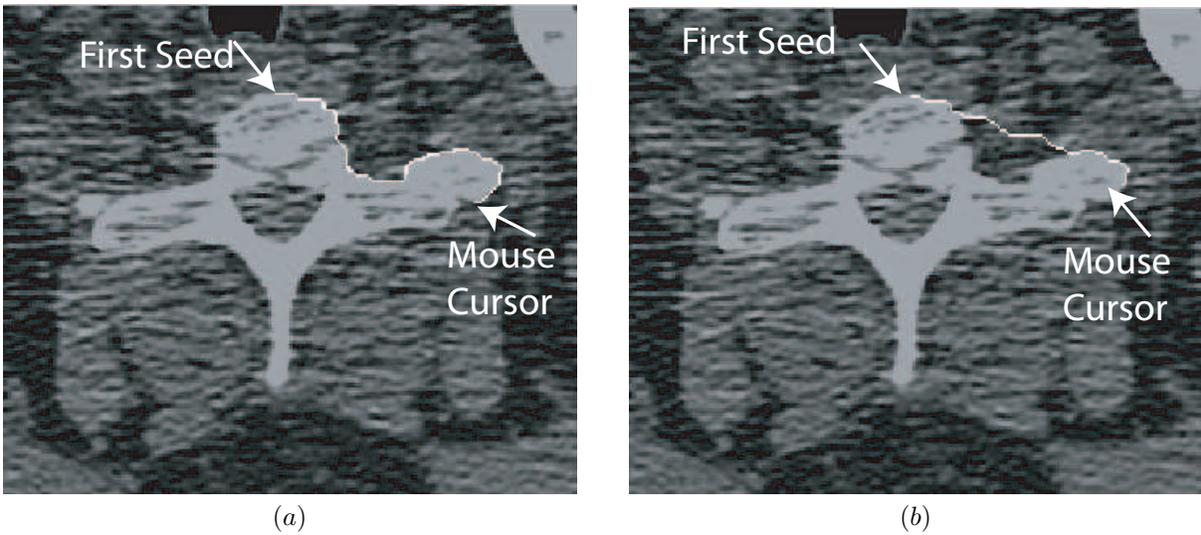


Figure 4: Illustration of the impact of new cost $f_{D_2}(p, r)$. (a) Result of the 2D live-wire graph search when $f_{D_2}(p, r)$ is included in cost $l(p, r)$ — a good boundary is found in a noisy image. (b) Result without $f_{D_2}(p, r)$ — the resulting boundary is poor. (Data used: 512×512 2D slice at $z = 0$ from human MDCT scan p2h052, generated by a Philips 4-detector MDCT scanner, with $\Delta x = \Delta y = 0.6\text{mm}$.)

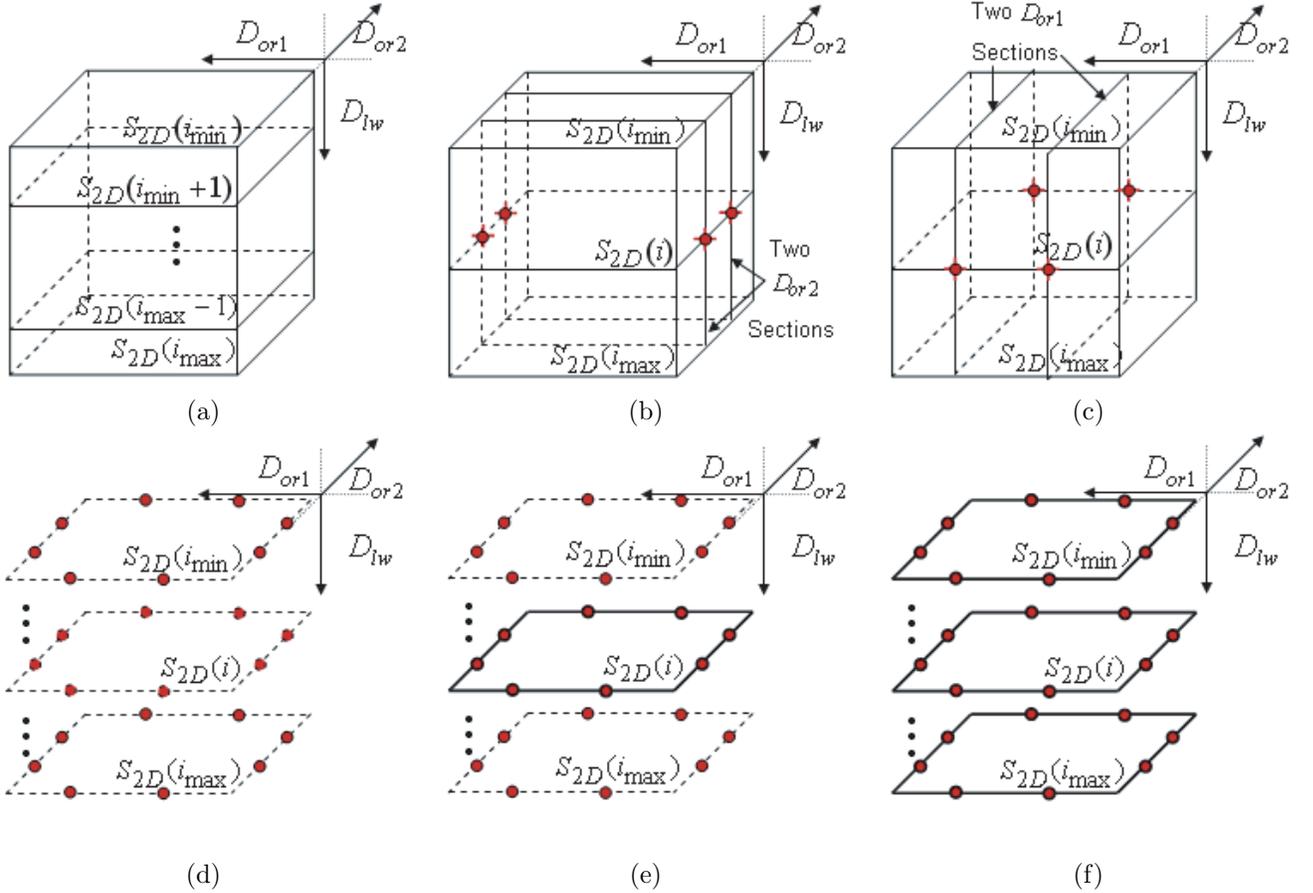


Figure 5: Schematic depiction of the new 3D live-wire method. (a) Study desired 3D ROI S , and pick 3D live-wire direction D_{lw} . This choice fixes the remaining orthogonal directions, D_{or1} and D_{or2} . S consists of the 2D boundaries $\{S_{2D}(i_{min}), S_{2D}(i_{min} + 1), \dots, S_{2D}(i_{max} - 1), S_{2D}(i_{max})\}$, where i_{min} and i_{max} are the smallest and largest section indices in direction D_{lw} for which S exists. In this schematic example, the ROI is the parallelepiped specified by the outlines. (b-c) On a few representative sections in one or both orthogonal directions, D_{or1} and D_{or2} , run the improved 2D live-wire method. In this example, two 2D boundaries were defined in each direction, D_{or1} and D_{or2} . (d) The four 2D boundaries just defined provide eight seeds for S on each section in direction D_{lw} . (e) Define a reference 2D boundary $S_{2D}(i)$ illustrated by a solid outline (and a working rectangle bounding the ROI) on section i in direction D_{lw} . (f) Run the automatic 3D live-wire process (Figure 6) to give the final segmented ROI S ; a complete 2D boundary exists on every section, $i_{min} \leq i \leq i_{max}$, in direction S_{lw} as shown by the solid outlines.

Inputs:
 r, B_r /* Reference section number and associated reference boundary */
 i_{min}, i_{max} /* Min and max section numbers for running automatic 3D live wire in direction D_{lw} */
 D_s /* Seed direction order—clockwise or counter-clockwise—from B_r */
 $\mathcal{S}(i)$ /* Seed set for section i (previously defined from sections in directions D_{or1} and D_{or2}) */

Data Structures:
 $L_s(i)$ /* List of seeds on section i */
 $B_{ref}(i)$ /* Reference object boundary for section i */
 $\text{Dist}(s, i)$ /* Distance between seed $s \in \mathcal{S}(i)$ and $B_{ref}(i)$ */
 T /* Distance threshold. Only seeds with $\text{Dist}(s, i) \leq T$ will be added into $L_s(i)$ */

Output
 $B_{obj}(i)$ /* Final ROI boundary for section i , $i_{min} \leq i \leq i_{max}$ */

Algorithm:
1. $i = r + 1$ /* Compute 2D boundaries for upper portion of ROI */
2. $B_{ref}(i) = B_r$ /* Use B_r as the reference section for section i */
3. while $i \leq i_{max}$ do
4. for each seed $s \in \mathcal{S}(i)$ do
5. if $\text{Dist}(s, i) \leq T$ then
6. $L_s(i) \leftarrow s$ /* Add seed s to seed list */
7. end
8. end
9. $\text{Order}(L_s(i), B_{ref}(i), D_s)$ /* Order seeds in $L_s(i)$ following order D_s */
10. $2\text{DLW}(L_s(i)) \rightarrow B_{obj}(i)$ /* Connect seeds in $L_s(i)$ using 2D live wire and output section i 's boundary */
11. $B_{ref}(i+1) = B_{obj}(i)$ /* Pass $B_{obj}(i)$ to the next section $i+1$ */
12. $i = i + 1$
13. end
14. $i = r - 1$ /* Compute 2D boundaries for lower portion of ROI */
15. $B_{ref}(i) = B_r$ /* Use B_r as the reference section for section i */
16. while $i \geq i_{min}$ do
17. for each seed $s \in \mathcal{S}(i)$ do
18. if $\text{Dist}(s, i) \leq T$ then
19. $L_s(i) \leftarrow s$ /* Add seed s into seed list */
20. end
21. end
22. $\text{Order}(L_s(i), B_{ref}(i), D_s)$ /* Order seeds in $L_s(i)$ following order D_s */
23. $2\text{DLW}(L_s(i)) \rightarrow B_{obj}(i)$ /* Connect seeds in $L_s(i)$ using 2D live wire and output section i 's boundary */
24. $B_{ref}(i-1) = B_{obj}(i)$ /* Pass $B_{obj}(i)$ to the next section $i-1$ */
25. $i = i - 1$
26. end

Figure 6: Automatic 3D live-wire algorithm. $\text{Order}(L_s(i), B_{ref}(i), D_s)$ is a function that orders the seeds in $L_s(i)$. $2\text{DLW}(L_s(i))$ denotes running the 2D live wire (Figure 2) on $L_s(i)$.

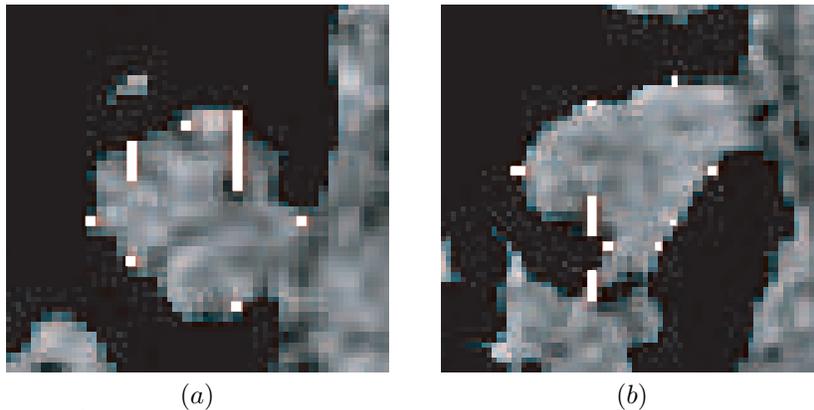


Figure 7: Examples of misleading seed circumstances on magnified sections in direction D_{lw} . (a) Incorrect seeds (white points) in the form of line segments from an orthogonal section processed in step 2 — many of these seeds are inside the correct boundary (Image p1h005, $z = 235$). (b) Incorrect seeds, again sometimes as line segments, exist outside the proper boundary (Image p1h005, $z = 257$).



Figure 8: Graphical user interface incorporating the proposed live wire methods. The white rectangle indicates a working rectangle specified by the user, while the enclosed white line indicates the beginning of an ROI's definition during 2D live wire. Refer to the text for more discussion.

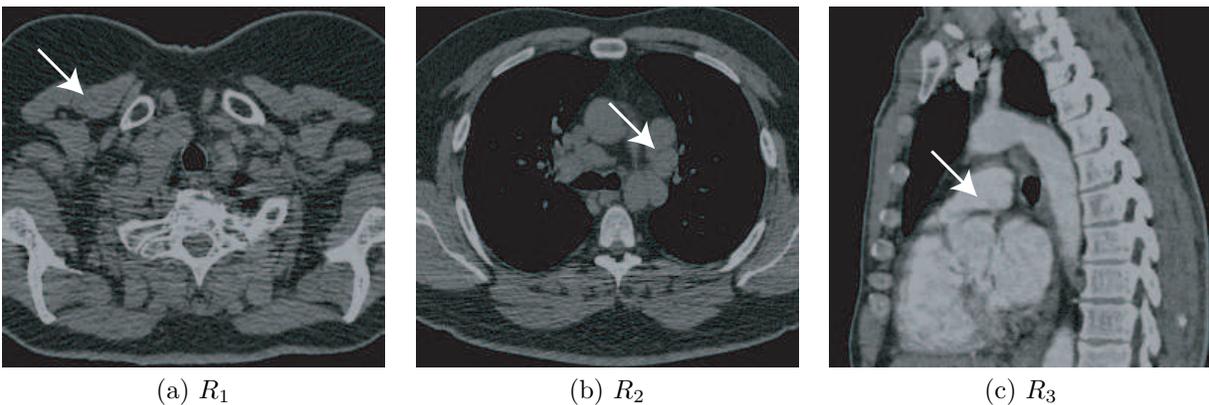
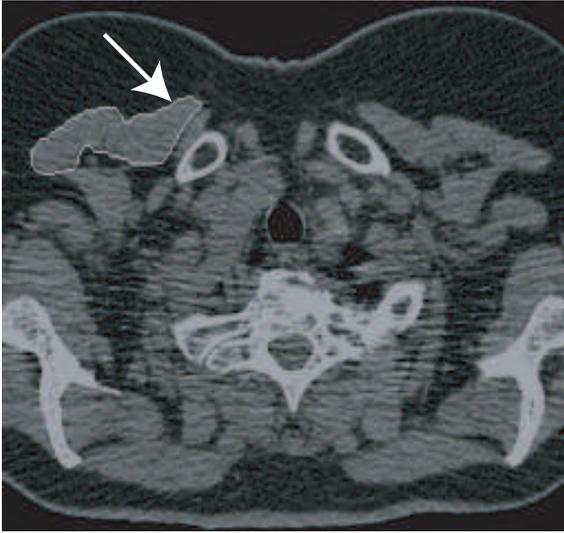
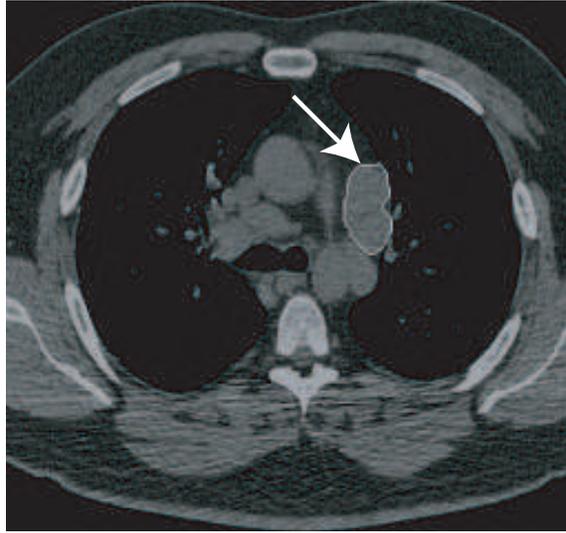


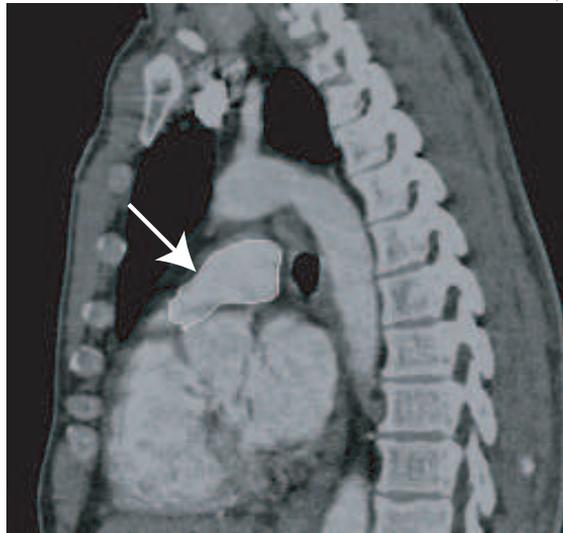
Figure 9: ROIs used for testing the improved 2D live-wire method; arrows point to ROIs in the figures (Table 1 gives scan details): (a) ROI R_1 , a transverse 2D section of human body above the right upper lung lobe; (b) ROI R_2 , a transverse 2D section of a subaortic lymph node; (c) ROI R_3 , a sagittal 2D section of the pulmonary artery close to the heart.



(a) R_1

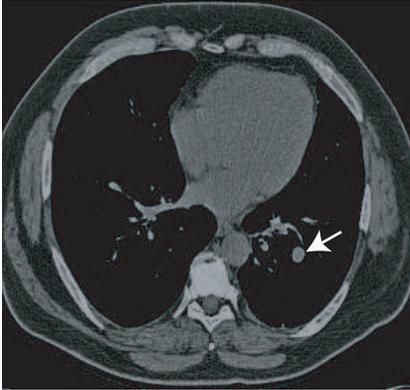


(b) R_2

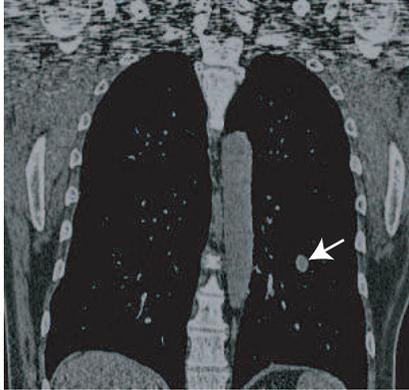


(c) R_3

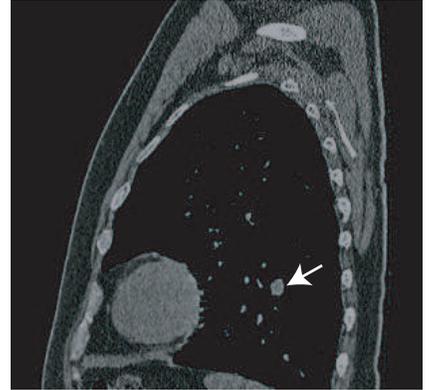
Figure 10: Example 2D live-wire segmentation results for the 2D ROIs.



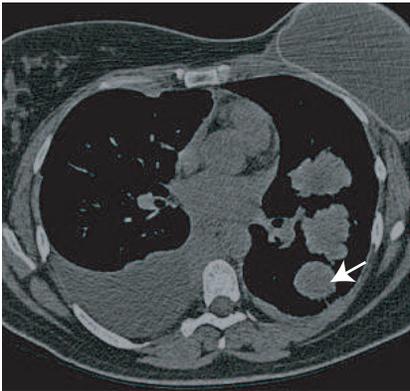
R_4 : transverse slice $z = 425$



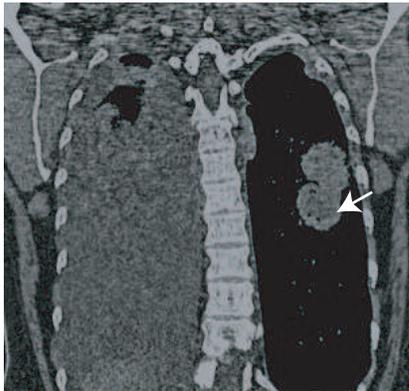
coronal section $y = 324$



sagittal section $x = 369$



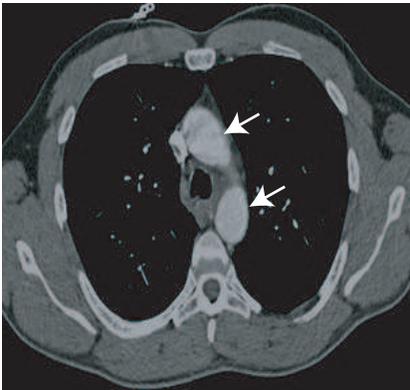
R_5 : 2D transverse slice $z = 226$



coronal section $y = 330$



sagittal section $x = 391$



R_6 : 2D transverse slice $z = 217$

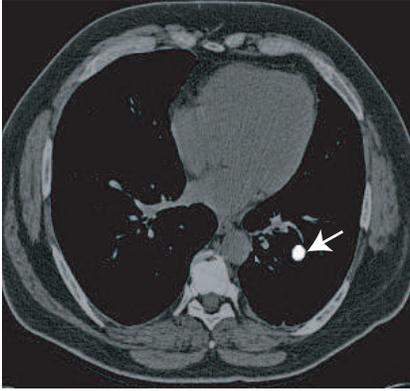


coronal section $y = 241$

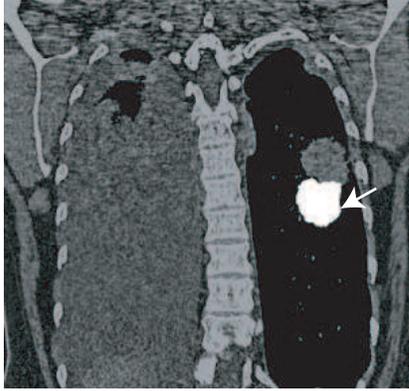


sagittal section $x = 274$

Figure 11: ROIs used for 3D segmentation tests; Table 4 gives scan details. Sample transverse, coronal, and sagittal sections are depicted for each ROI, with arrows indicating ROI positions. Row 1: R_4 . Row 2: R_5 . Row 3: R_6 .



R_4 : transverse slice $z = 425$

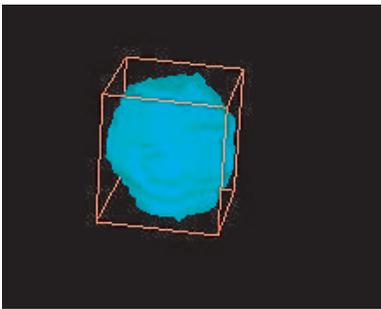


R_5 : coronal section $y = 330$

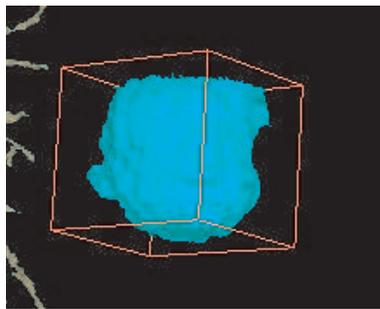


R_6 : coronal section $y = 241$

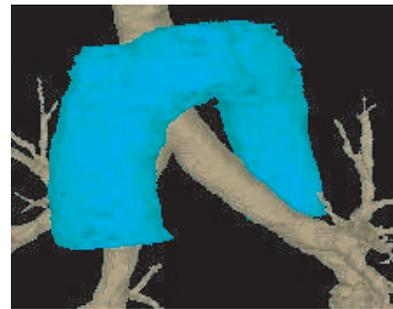
Figure 12: 3D live-wire segmentations for the 3D ROIs. Sample sections for R_4 , R_5 , and R_6 .



(j)



(k)



(l)

Figure 13: Segmentation result of 3D ROIs in three viewing directions. (a), (b), (c): 3D renderings (cyan) for ROI R_4 , R_5 , and R_6 .

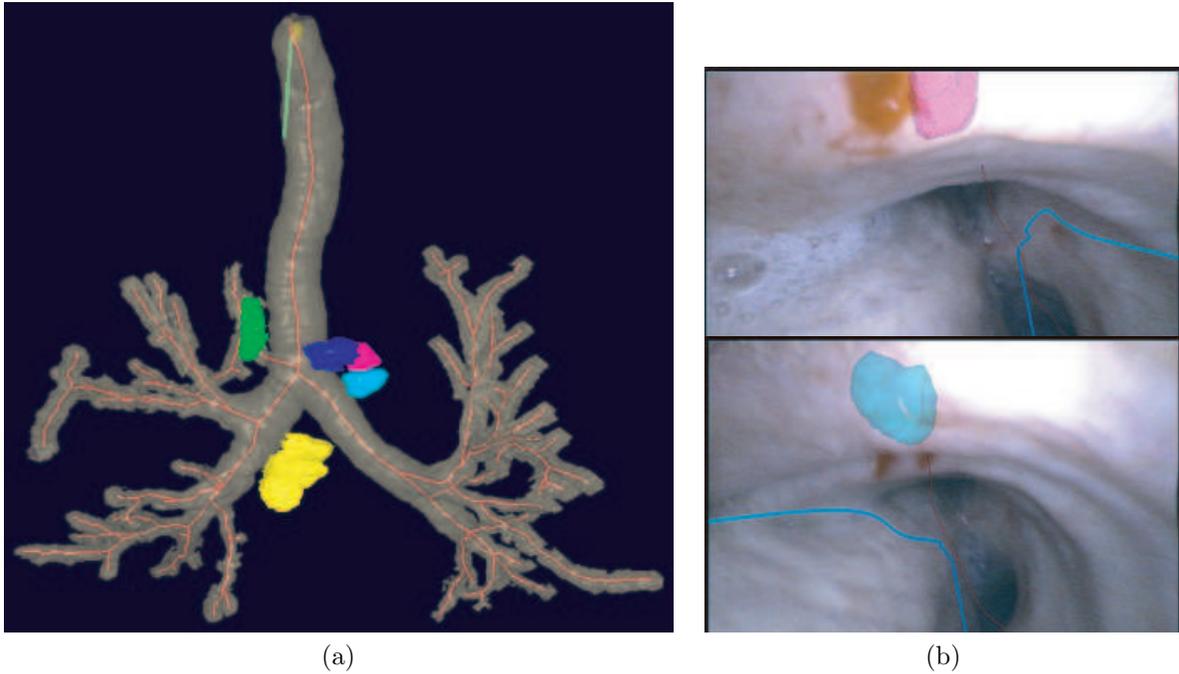


Figure 14: Sample views during live MDCT-based guidance of bronchoscopy for a human lung-cancer patient. (a) Surface rendering of segmented airway tree, mediastinal lymph nodes, and airway centerlines (red). Before bronchoscopy, the physician used the 3D live-wire method to define the lymph nodes of interest: green — right paratracheal node; blue, magenta, light blue — left paratracheal nodes; yellow — subcarinal node. (Patient exhibits a rare “pig bronchus”; hence, the trifurcation at the main carina.) (b) Live bronchoscopic video views during the procedure showing an MDCT-defined ROI and centerline fused onto the registered video view: top — magenta left paratracheal node; bottom — light-blue left paratracheal node. The ROIs, not visible in the live video, are now observable to the physician in the live bronchoscopic video. (Case 20349_3.3: $512 \times 512 \times 577$ 3D MDCT scan generated on a Siemens Sensation-16 scanner; $\Delta x = \Delta y = 0.715\text{mm}$, $\Delta z = 0.5\text{mm}$. Airway tree segmented via the methods of [4] and the centerlines were subsequently defined using the method of [8, 42, 43].)