

# Efficient Optimal Surface Detection: Theory, Implementation and Experimental Validation

Kang Li<sup>a</sup>, Xiaodong Wu<sup>b</sup>, Danny Z. Chen<sup>c</sup>, Milan Sonka<sup>a</sup>

<sup>a</sup>The University of Iowa, Dept. of Electrical and Computer Eng., Iowa City, IA, USA

<sup>b</sup>The University of Texas – Pan American, Dept. of Computer Science, Edinburg, TX, USA

<sup>c</sup>The University of Notre Dame, Dept. of Computer Science and Eng., Notre Dame, IN, USA

## ABSTRACT

In this paper, a novel polynomial-time algorithm is described for solving the *optimal net surface detection* problem on *proper ordered multi-column* graphs in  $N$ -D space ( $N \geq 3$ ). The method is applied to searching for optimal object boundaries with arbitrary smoothness constraints in volumetric medical images. By simple transformations, such optimal surface detection problems can be simplified to a problem of computing the minimum  $s$ - $t$  cuts in the transformed graphs. An efficient implementation for the 3-D case that can achieve near real-time performance on moderate-sized datasets is presented. We further examine our technique in experiments by segmenting the cylindrical surfaces of human airways from pulmonary volumetric CT images, and compare the results to those produced by previous methods. By allowing full specifications of the cost-function and smoothness constraints without degrading the performance, the new algorithm is more flexible than traditional methods and guarantees global optimality. The multi-dimensional nature of the algorithm maintains continuity in higher dimensions.

**Keywords:** Optimal Surface, Surface Segmentation, Minimum  $s$ - $t$  Cut, Graph Algorithm, CT

## 1. INTRODUCTION

The task of identifying globally optimal three-dimensional surfaces representing object boundaries is common in segmentation of volumetric medical images. The accuracy of the segmentation has great impact on the effectiveness of the subsequent quantitative measurements and other processing steps. Many existing segmentation techniques either do not consider global optimality at all, or compute only an approximate solution that may be arbitrarily far away from the global optimum (level set methods, surface growing [1], fuzzy connectedness [2,3], etc). There were a few attempts to achieve optimal surface detection based on, for instance, cost minimization using graph searching principles [4]. While such algorithms guarantee surface optimality, they are impractical due to their enormous computational requirements, which are usually exponential in terms of the data sizes of the problems.

In this paper, we present a new and efficient algorithm for detecting optimal surfaces with given smoothness constraints in  $N$ -dimensional images ( $N \geq 3$ ) [5]. We present both the generic theoretical setup and an efficient implementation for the 3-D case. The underlying theory of the graph algorithm was developed very recently [5] and had never been implemented as a computer program or applied to medical image segmentation problems before.

The algorithm was validated in many computer simulated images as well as in a practical application of segmenting the inside wall surfaces of intrathoracic airways imaged by multi-detector CT. The theoretical advantages of the algorithm were proved by experiments. Compared to the 2-D slice-by-slice approaches, the new algorithm was able to produce more accurate results with comparable efficiency.

---

Contact information: milan-sonka@uiowa.edu

## 2. PROBLEM MODELING

A volumetric dataset can be viewed as a 3-D matrix  $\mathcal{I}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ . Without loss of generality (WLOG), the desired surface is considered terrain-like and oriented as shown in Figure 1 (a). Let  $X$ ,  $Y$  and  $Z$  denote the image sizes in  $\mathbf{x}$ ,  $\mathbf{y}$  and  $\mathbf{z}$  directions, respectively. A requirement is that the desired surface intersects with exactly one voxel of each *column* parallel to the  $\mathbf{z}$ -axis. By defining a cost function, a cost value is computed and associated with each voxel  $\mathcal{I}(x, y, z)$  of  $\mathcal{I}$ , denoted  $c(x, y, z)$ . Generally, the cost value is inversely related to the likelihood that the desired surface would contain the voxel  $\mathcal{I}(x, y, z)$ . An *optimal surface* is the one with the minimum total sum of voxel costs among all feasible surfaces that can be defined in the 3-D volume. The feasibility of a surface is constrained by the application specific *smoothness parameters*,  $\Delta_{\mathbf{x}}$  and  $\Delta_{\mathbf{y}}$ , guaranteeing surface continuity in 3-D. More precisely, If  $\mathcal{I}(x, y, z)$  and  $\mathcal{I}(x + 1, y, z')$  are two voxels on a feasible surface, then  $|z - z'| \leq \Delta_{\mathbf{x}}$ . Likewise, if  $\mathcal{I}(x, y, z)$  and  $\mathcal{I}(x, y + 1, z')$  are two voxels on a feasible surface, then  $|z - z'| \leq \Delta_{\mathbf{y}}$ .

A weighted directed graph  $G = (V, E)$  is constructed according to  $\mathcal{I}$  as follows. Every vertex  $V(x, y, z) \in V$  represents one and only one voxel  $\mathcal{I}(x, y, z) \in \mathcal{I}$ , whose cost  $w(x, y, z)$  is assigned according to:

$$w(x, y, z) = \begin{cases} c(x, y, z) & \text{if } z = 0 \\ c(x, y, z) - c(x, y, z - 1) & \text{if } z > 0 \end{cases} \quad (1)$$

For each  $(x, y)$  pair satisfying  $0 \leq x < X$  and  $0 \leq y < Y$ , the vertex subset  $\{V(x, y, z) : 0 \leq z < Z\}$  is called the  $(x, y)$ -*column* of  $G$ , denoted by  $Col(x, y)$ . Along each column  $Col(x, y)$ , every vertex  $V(x, y, z) (z > 0)$  has a directed edge to the vertex  $V(x, y, z - 1)$ , denoted  $(V(x, y, z), V(x, y, z - 1))$ .

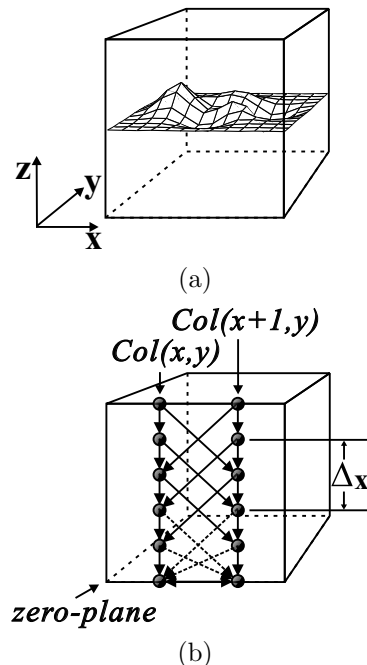
The smoothness constraints along the  $\mathbf{x}$ - and  $\mathbf{y}$ -directions are enforced in the following way. WLOG, 4-neighbor adjacency is assumed. The construction described below can be easily extended to other adjacency settings. Consider any two adjacent columns,  $Col(x, y)$  and  $Col(x + 1, y)$  as shown in Figure 1 (b). Along the  $\mathbf{x}$ -direction and for any  $0 \leq x < X - 1$ , a directed edge is constructed from each vertex  $V(x, y, z) \in Col(x, y)$  (resp.,  $V(x + 1, y, z) \in Col(x + 1, y)$ ) to vertex  $V(x + 1, y, \max(0, z - \Delta_{\mathbf{x}})) \in Col(x + 1, y)$  (resp.,  $V(x, y, \max(0, z - \Delta_{\mathbf{x}})) \in Col(x, y)$ ). The same construction is done for the  $\mathbf{y}$ -direction.

Sometimes the target surface  $S$  is required to be *wraparound* along the  $\mathbf{x}$ - (or  $\mathbf{y}$ -) direction. This is common when we are to detect a cylindrical surface, which is required to be unfolded to a terrain-like surface using cylindrical coordinate transform [6] before applying our algorithm (or the traditional graph-searching algorithms [1, 4, 7]). Then the first and last rows along the unfolding plane should satisfy the smoothness constraints as well. In the  $\mathbf{x}$ -wraparound case, each vertex  $V(0, y, z)$  (resp.,  $V(X - 1, y, z)$ ) also connects to  $V(X - 1, y, \max(0, z - \Delta_{\mathbf{x}}))$  (resp.,  $V(0, y, \max(0, z - \Delta_{\mathbf{x}}))$ ). The same rule applies to the  $\mathbf{y}$ -wraparound case.

## 3. THE ALGORITHM

### 3.1. The Minimum Closed Set

In Section 2, the construction of a weighted directed graph  $G = (V, E)$  from the volumetric dataset  $\mathcal{I}(\mathbf{x}, \mathbf{y}, \mathbf{z})$  was described. Actually, the goal of such a construction is to guarantee that: 1) any feasible surface in  $\mathcal{I}$  correspond to a non-empty closed set in  $G$  with the same total cost; 2) any non-empty closed set in  $G$  defines a feasible surface in  $\mathcal{I}$  with the same total cost. Notice that in  $G$ , if  $V(x, y, z)$  is in a closed set  $\mathcal{C}$



**Figure 1.** Modeling of the optimal surface detection problem. (a) The surface orientation. (b) Two adjacent columns in the constructed graph. The smoothness parameter  $\Delta_{\mathbf{x}} = 2$ .

of  $G$ , then all vertices “below” it on  $Col(x, y)$  are in  $\mathcal{C}$ . We denote the set of these vertices  $B^z(x, y)$ , i.e.,  $B^z(x, y) = \{V(x, y, z') : z' \leq z\}$ . The cost of  $B^z(x, y)$  equals the cost of vertex  $V(x, y, z)$  in  $G$ . In general, we are able to prove the following lemma, showing that computing the optimal surface in  $\mathcal{I}$  is equivalent to finding a non-empty closed set  $\mathcal{C}^*$  of the minimum cost in  $G$ . The proof of this lemma was given in [5].

LEMMA 3.1. *A non-empty minimum closed set  $\mathcal{C}^*$  in  $G$  specifies the optimal surface in  $\mathcal{I}$ .*

However, if a closed set  $\mathcal{C}^*$  of the minimum cost (called *minimum closed set*) in  $G$  is empty,  $\mathcal{C}^*$  gives little useful information on  $G$  to define the optimal  $k$  surfaces in  $\mathcal{I}$ . This problem can be easily solved by performing a *translation* operation on  $G$  as introduced in [5]. After the translation, we can simply find a minimum closed set  $\mathcal{C}^*$  in  $G$ , and  $\mathcal{C}^*$  is a minimum *non-empty* closed set in  $G$  before the translation.

Note that the directed edges connecting vertices not on the zero-plane to the vertices on the zero-plane (shown in dashed lines in Figure 1 (b)) are optional. The vertices of  $G$  on the zero-plane form a *strongly connected component*, as is ensured by the construction, and they are all included in any non-empty closed set of  $G$ . Therefore, removing these edges does not affect any closed set in  $G$ . This gives rise to an very interesting observation: the graph is actually getting simpler (i.e., with less edges) as the smoothness constraints are relaxed (i.e.,  $\Delta_x$  and/or  $\Delta_y$  become larger). This behavior is just the opposite to the traditional graph-search based algorithms for the problem.

### 3.2. Computing the Optimal Surface

Based on Lemma 3.1, we need to compute a minimum-cost *non-empty* closed set  $\mathcal{C}^*$  in  $G$ , which is a well studied problem in graph theory. As in [5, 8, 9], we compute  $\mathcal{C}^*$  in  $G$  by further transforming it into computing a minimum  $s$ - $t$  cut in a related graph  $G_{st}$ .

Let  $V^+$  and  $V^-$  denote the set of vertices in  $G$  with non-negative and negative costs, respectively. Define a new digraph  $G_{st} = (V \cup \{s, t\}, E \cup E_{st})$ . The vertex set of  $G_{st}$  is the vertex set  $V$  of  $G$  plus a source  $s$  and a sink  $t$ . The edge set of  $G_{st}$  is the edge set  $E$  of  $G$  plus a new edge set  $E_{st}$ . We assign an infinity ( $+\infty$ ) weight to each edge in  $E$ .  $E_{st}$  consists of the following edges: The source  $s$  is connected to each vertex  $v \in V^-$  by a directed edge of weight  $-w(v)$ , and every vertex  $v \in V^+$  is connected to the sink  $t$  by a directed edge of weight  $w(v)$ , where  $w(v)$  is the cost of vertex  $v$ . An  $s$ - $t$  cut in  $G_{st}$  is a subset of edges whose removal disconnects  $t$  from  $s$  (i.e., there will be no paths from  $s$  to  $t$ ), and it partitions the vertex set into two subsets  $\mathcal{S}$  and  $\bar{\mathcal{S}}$  with  $s \in \mathcal{S}$  and  $t \in \bar{\mathcal{S}}$ . Let  $(\mathcal{S}, \bar{\mathcal{S}})$  denote a finite-weight  $s$ - $t$  cut in  $G_{st}$ , and  $w(\mathcal{S}, \bar{\mathcal{S}})$  denote the total weight of the cut. It was proven that

$$w(\mathcal{S}, \bar{\mathcal{S}}) = -w(V^-) + \sum_{v \in \mathcal{S} - \{s\}} w(v) \quad (2)$$

where  $-w(V^-)$  is fixed and is the cost sum of all vertices with negative costs in  $G$ . Since  $\mathcal{S} - \{s\}$  is a closed set in  $G$  [8, 9], the weight of a cut  $(\mathcal{S}, \bar{\mathcal{S}})$  in  $G_{st}$  and the cost of the corresponding closed set in  $G$  differ by a constant. Hence, the source set  $\mathcal{S}$  of a minimum cut in  $G_{st}$  corresponds to a minimum closed set  $\mathcal{C}^*$  in  $G$ .

Note that if the graph  $G_{st}$  has  $O(kn)$  vertices and  $O(kn)$  edges, then the minimum closed set  $\mathcal{C}^*$  in  $G$  can be computed in  $T(kn, kn)$  time, herein  $T(kn, kn)$  is the time for finding a minimum  $s$ - $t$  cut in an edge-weighted directed graph with  $O(kn)$  vertices and  $O(kn)$  edges.

We recover the optimal surface based on  $\mathcal{C}^*$  in the following way. For every  $0 \leq x < X$  and  $0 \leq y < Y$ , let  $B(x, y) = \mathcal{C}^* \cap Col(x, y)$ . Denote by  $V(x, y, z^*)$  the vertex in  $B(x, y)$  with the largest  $\mathbf{z}$ -coordinate. Then, voxel  $\mathcal{I}(x, y, z^*)$  is on the optimal surface  $\mathcal{N}^*$ . In this way, the minimum closed set  $\mathcal{C}^*$  of  $G$  defines the optimal surface  $\mathcal{N}^*$  in  $\mathcal{I}$ .

To sum up, we have the following result.

THEOREM 3.2. *The optimal surface in a 3-D image  $\mathcal{I}(\mathbf{x}, \mathbf{y}, \mathbf{z})$  with  $n$  voxels can be computed in  $T(kn, kn)$  time.*

## 4. EFFICIENT IMPLEMENTATION

As discussed in the previous sections, we can divide the computation for finding the optimal surface in a volumetric image into three major steps: (1) construct the transformed graph  $G_{st}$ ; (2) compute the minimum  $s-t$  cut in  $G_{st}$ ; and (3) recover the optimal surface from the source set of the computed minimum  $s-t$  cut. The first and last steps are straightforward, and thus we will mainly focus on the second step: computing the minimum  $s-t$  cut. The minimum  $s-t$  cut problem and its dual, the *maximum flow* problem, are classic combinatorial problems that can be solved by various polynomial-time algorithms. The most commonly used algorithms can be classified into two categories: the *augmenting path* family and the *push-relabel* family. Though the push-relabel algorithm is commonly considered to be the most efficient for many applied problems, it was shown by Boykov and Kolmogorov [10] that for a class of  $s-t$  graphs frequently encountered in computer vision problems, the improved algorithm they devised based on the augmenting path method is more efficient in practice. In the context of our surface detection problem, we have compared the running time of the Boykov-Kolmogorov (BK) algorithm, the Edmonds-Karp augmenting path algorithm, and the push-relabel algorithm in experiments. The BK algorithm was shown to be the fastest for our purpose. Thus, we have chosen the BK algorithm as a basis of our implementation.

Memory efficiency is another key problem in practical multidimensional image processing. Choosing an appropriate graph representation has therefore become an important factor. Since the minimum  $s-t$  cut algorithm operates on the *residual graph* of  $G_{st}$ , where for each edge in  $G_{st}$ , its reverse edge is also allocated. Therefore, the total number of edges is actually doubled in the residual graph. It turned out that many common graph representations (“adjacency list”, “forward star”) are not ideal for representing the graph we need. In medical image processing where large data sets are ubiquitous, this limitation is hindering. Fortunately, this issue can be fairly easily resolved by utilizing the high regularity of the graph  $G_{st}$  we used. Noticing that  $G_{st}$  consists of several sets of parallel edges and the number of outgoing edges for each vertex is known once the smoothness and surface separation parameters are given, we can represent all edges in  $E_{st}$  *implicitly* by simply storing several offset values. Note also that the edges incident to the source and sink can always be implicitly represented, since they are only auxiliary. Thus, no edge in  $G_{st}$  needs to be explicitly constructed. This *implicit-edge* scheme saved more than 50% of memory compared to the “forward-star” representation.

## 5. EXPERIMENTS

### 5.1. Data

The proposed algorithm was examined on two different kinds of volumetric datasets: computer generated phantoms and in vivo acquired human pulmonary CT data.

To evaluate the execution times and memory usages of different implementations of the algorithm, a set of computer phantoms embedded with terrain-like surfaces were used. The sizes ( $X \times Y \times Z$ ) of these phantoms ranged from  $20 \times 20 \times 20$  to  $300 \times 300 \times 300$ . All computer-generated phantoms used the 32-bit floating point data type to store voxel brightness values. The voxels at the desired surface positions were assigned lower brightness values compared to the background. The datasets were blurred and superimposed with slight Gaussian noise to make the problem more realistic.

To demonstrate the utility of our method in quantitative analysis of human pulmonary CT images, the algorithm was incorporated into an automated system for pulmonary airway analysis [11]. The inside wall surfaces of the intrathoracic airways were determined in 12 in vivo CT scans of 6 human subjects. For each subject, a scan close to total lung capacity (TLC) was acquired (at 85% lung volume), and a scan close to functional residual capacity (FRC) was acquired (at 55% lung volume). All in vivo scans have a nearly isometric resolution of  $0.7 \times 0.7 \times 0.6 \text{ mm}^3$  and consist of 500-600 image slices,  $512 \times 512$  pixels each.

### 5.2. Segmentation of Airway Wall Surfaces

The in vivo airway walls were segmented and analyzed using an automated system designed for pulmonary airway analysis [11]. The entire segmentation and analysis process was carried out with no human intervention

and consisted of the following steps: 1) topological “pre-segmentation” of the airway tree; 2) airway tree skeletonization and labeling; 3) 3-D data resampling; 4) airway-wall segmentation (using the reported algorithm), and 5) quantitative measurement.

At first, the entire airway tree was segmented from the pulmonary CT dataset using a multi-seed fuzzy-connectedness technique [11,12]. This step is called a “pre-segmentation” since it gives the approximate spatial locations of the individual airway segments; however, accurate positioning of the airway boundaries is not guaranteed. The centerlines of the airway branches were obtained by applying a skeletonization algorithm [13, 14]. Meanwhile, each branch point of the airway was assigned a unique anatomic label. Following the centerline, each airway segment between two branch points, excluding the branching parts, was resampled using the cubic B-spline interpolation technique [15,16] so that the slices in the resampled volumes are always perpendicular to the centerlines. After this step, about 30 resampled airway segments can be obtained from each CT dataset. The resampled volumes were input to our algorithm, by which the precise inside airway walls were segmented, producing the spatial coordinates of wall surfaces. Since the airway wall surface is tubular, the resampled volumes were unfolded before applying the optimal surface detection algorithm. The size of the unfolded image was adaptively determined according to the average estimated inside wall diameter of each airway segment, which was computed from the pre-segmented airway tree. The cost function used for the segmentation utilized a combination of first and second derivatives of 2-D gray-level images [17]. The segmented wall surface was smoothed slice-by-slice by applying an averaging filter. Finally, quantitative measurements were performed in each segmented slice: the major and minor wall diameters and the cross-section area enclosed by the airway-wall were obtained.

To reflect the shapes of the airway walls and the near-isometric character of the in vivo pulmonary CT images, the values of intra-slice and inter-slice smoothness parameters were both set to 3 for the airway wall segmentation using our algorithm. For the segmentation of phantom plexiglas tubes, an intra-slice smoothness of 1 and an inter-slice smoothness of 2 were used.

### 5.3. Performance Indices and Data Analysis

The performance of the reported algorithm was evaluated in three different ways.

First, the execution times were recorded and compared in 66 computer-generated phantoms of varying sizes to gain a basic understanding of the speed/size relationships. The execution time included the graph initialization time and the actual computation time. Three implementations of the algorithm were tested, which used three different graph representations: “adjacency-list” (AL), “forward-star” (FS) and “implicit-edge” (IE). All experiments were conducted on an AMD Athlon MP 2000+ (1.67GHz) Dual CPU workstation with 3.5 GB of memory running Microsoft Windows. The execution time of each implementation was measured three times and the results were averaged.

Second, along with the execution times, the peak memory consumptions of our testing programs were also recorded. This information was gathered by calling the Windows API function “`GetProcessMemoryInfo`” at the end of the testing programs. The peak memory consumption reflects the total memory consumed by the testing dataset and the specific implementation of our algorithm. It corresponds to the “peak working set size” value returned by the API function call, which is the maximum amount of memory physically mapped to the testing program’s process context during its execution [18].

Third, the previous 2-D segmentation approach [11] was qualitatively compared with the new 3-D method in in vivo CT data. A total of 317 airway segments consisting of 4212 perpendicular cross-sectional image slices were automatically segmented and analyzed by computer. An independent expert observer visually inspected the airway wall segmentation results in 100 randomly selected airway segments. The expert was supplied with the segmentation results in pairs. In each pair, one result was obtained using the traditional slice-by-slice dynamic programming method, and the other was obtained by our new approach. The expert evaluated the detected wall in the result pairs by answering a single-choice question, unaware of the underlying segmentation method used. The single-choice question consisted of five different answers: “the results are good and equivalent”, “result 1 is better”, “result 2 is better”, “both failed” and “uncertain”. The results were considered good and equivalent if the detected borders did not substantially differ from each other in any

single one of the cross-sectional slices forming the airway segment. Otherwise, the boundaries more closely representing the true boundaries were considered superior. The expert could choose “both failed” if none of the methods correctly represented the true wall boundaries in all slices of the segment, or “uncertain” if the true wall boundaries were difficult to visually locate due to bad image quality.

## 6. RESULTS

### 6.1. Execution Time and Memory Usage

The measured execution times and peak memory consumptions of the three implementations of our algorithm are shown in Table 1. The data given in the “AL”, “FS” and “IE” columns are the results of the “adjacency-list”, “forward-star” and “implicit-edge” implementations, respectively. The image sizes are shown in voxels. The symbol “–” represents unsuccessful measurements, where our programs were terminated by the operating system because their peak memory consumptions exceeded 2 Gigabytes. The dramatic speed and memory usage improvement of the “implicit-edge” implementation over the previous two data structures is clearly reflected by the results.

**Table 1.** Execution Time and Memory Usage

Image Size	Execution Time (sec.)			Peak Mem. Usage (MB)		
	AL	FS	IE	AL	FS	IE
$50 \times 50 \times 50$	0.36	0.56	0.17	20.63	13.11	6.62
$100 \times 100 \times 100$	24.17	17.20	5.31	169.94	102.88	44.34
$150 \times 150 \times 150$	130.06	101.67	54.83	585.93	351.82	146.61
$200 \times 200 \times 200$	411.03	328.34	185.89	1405.42	841.24	345.68
$250 \times 250 \times 250$	–	770.99	463.94	–	1652.42	673.80
$300 \times 300 \times 300$	–	–	957.17	–	–	1163.24

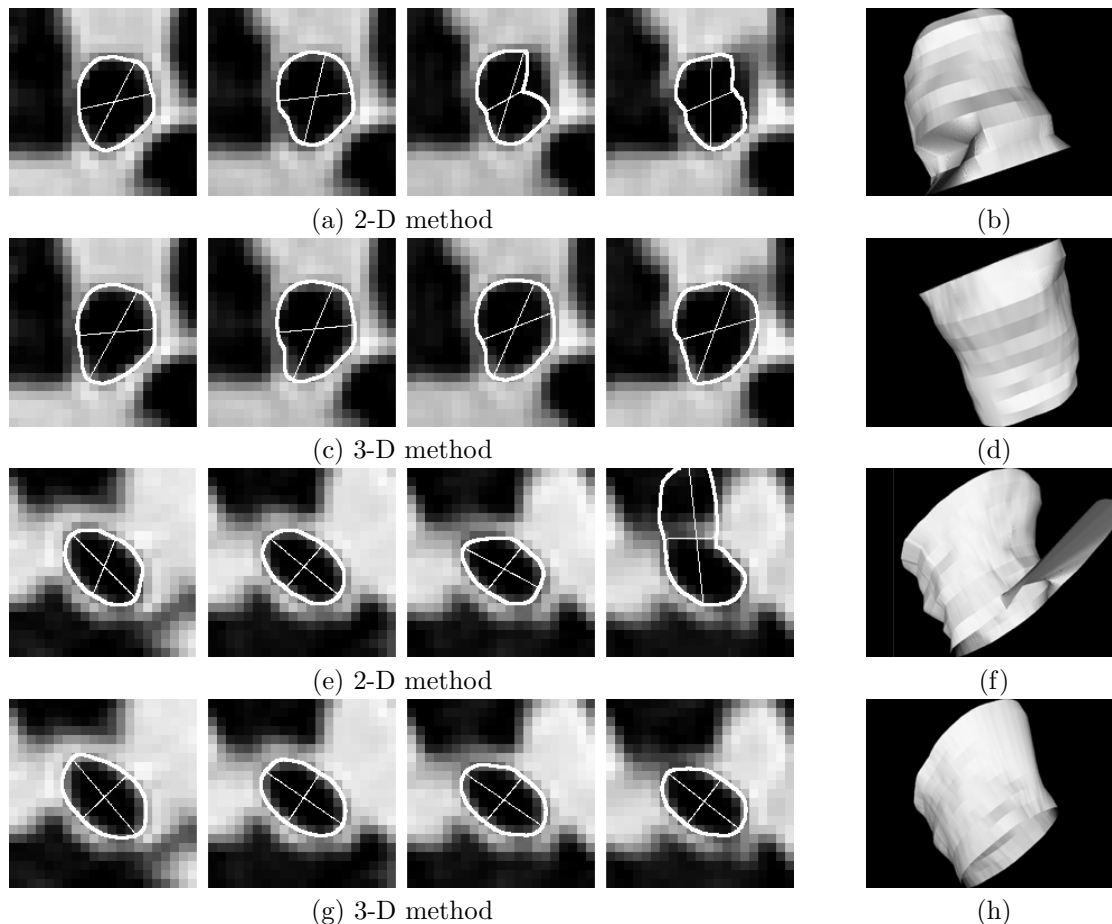
### 6.2. Airway Wall Segmentation

Segmenting and measuring one entire airway tree (about 30 airway segments) took approximately 3.5 minutes excluding the time used for pre-segmentation and skeletonization. Expert assessments of the segmentation results were as follows. Among the 100 randomly selected airway segments, our optimal surface detection technique yielded superior results to the 2-D approach in 45 segments. 52 result pairs were considered equivalent. On 1 airway segment both methods failed, and the expert was uncertain about the remaining 2 cases. Samples of the segmentation results are presented in Figure 2. Typically, benefiting from the incorporation of 3-D connectedness information, the 3-D optimal surface detection method was able to follow the airway wall structures correctly even in small, blurred airway segments, where the 2-D method exhibited large perturbation, or was even completely lost in some image slices.

## 7. DISCUSSION

Like the traditional graph searching based methods, the proposed approach has two apparent limitations. One is that it can only detect those surfaces that can be “unfolded” to be terrain-like, including cylindrical or tubular surfaces, and this unfolding process must be invertible. Closed surfaces, such as spherical ones, are not subject of our algorithm since there is no perfectly invertible method to unfold them. In addition, the current approach cannot handle topology changes. An example of such a limitation is the inability to directly segment branching tubular structures.

Since the segmentation problem was transformed into solving a minimum  $s$ - $t$  cut in directed 3-D geometric graphs, the performance of the approach crucially hinges on the minimum  $s$ - $t$  cut algorithm being used. Our current implementations used the minimum  $s$ - $t$  cut algorithm reported in [10]. Further improvements can



**Figure 2.** Comparison of airway wall segmentation results. (a), (b), (e) and (f) are the results yielded by the slice-by-slice dynamic programming approach in 2 different airway segments. 4 consecutive slices and the 3-D surface rendering are shown for each airway segment. (c), (d), (g) and (h) are the walls segmented by the reported 3-D approach.

be made by adopting more sophisticated minimum  $s$ - $t$  cut/maximum flow algorithms (for example, see [19]). Another option is to parallelize the algorithm. Preliminary results show that simple parallelizing techniques (e.g., multi-threading) can significantly accelerate our program, especially on large datasets. Formal parallel minimum  $s$ - $t$  cut algorithms are described in literature, e.g., [20,21]. The multi-scale approach commonly used in image processing provides yet another alternative to speed up our method.

## 8. CONCLUSION

A polynomial-time algorithm for detecting the optimal surface in volumetric images was developed and validated in a medical image segmentation problem. The method performs efficiently on moderate-sized datasets and is robust in the sense that the resulting surfaces are globally optimal with respect to the employed cost functions and smoothness constraints. The multidimensional nature of the algorithm ensures 3-D consistency of the results. The surface smoothness parameters provide a flexible and mathematically justified means for modeling various inherent properties of the desired surfaces. The methods are readily extendible to higher dimensions.

## ACKNOWLEDGMENTS

The authors thank Dr. Juerg Tschirren for providing experimental data, algorithm testing environment, and technical support. CT imaging was performed under guidance of Drs. Eric A. Hoffman and Geoffrey McLennan. The research was supported in part by NIH grants R01-HL063373 and R01-HL071809.

## REFERENCES

1. R. J. Frank, "Optimal surface detection using multi-dimensional graph search: Applications to intravascular ultrasound," Master's thesis, The University of Iowa, 1996.
2. J. K. Udupa and S. Samarasekera, "Fuzzy Connectedness and Object Definition: Theory, Algorithms, and Applications in Image Segmentation," *Graphics Models and Image Processing* **58**, pp. 246–261, 5 1996.
3. G. T. Herman and B. M. Carvalho, "Multiseeded Segmentation Using Fuzzy Connectedness," *IEEE Trans. Pattern Anal. Machine Intell.* **23**, pp. 460–474, 5 2001.
4. D. R. Thedens, D. J. Skorton, and S. R. Fleagle, "A three-dimensional graph searching technique for cardiac border detection in sequential images and its application to magnetic resonance image data," in *Computers in Cardiology*, pp. 57–60, IEEE, (Los Alamitos, CA), 1990.
5. X. Wu and D. Z. Chen, "Optimal Net Surface Problems with Applications," in *Proc. of the 29th International Colloquium on Automata, Languages and Programming (ICALP)*, pp. 1029–1042, (Malaga, Spain), July 2002.
6. M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, Brooks/Cole Publishing Company, second ed., 1999.
7. D. R. Thedens, D. J. Skorton, and S. R. Fleagle, "Methods of graph searching for border detection in image sequences with application to cardiac magnetic resonance imaging," *IEEE Trans. Med. Imaging* **14**, pp. 42–55, 1995.
8. J. C. Picard, "Maximal Closure of a Graph and Applications to Combinatorial Problems," *Management Science* **22**, pp. 1268–1272, 1976.
9. D. S. Hochbaum, "A New-Old Algorithm for Minimum-Cut and Maximum-Flow in Closure Graphs," *Networks* **37**, pp. 171–193, 2001.
10. Y. Boykov and V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Computer Vision," in *Third International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, (Springer-Verlag), Sept. 2001.
11. J. Tschirren, *Segmentation, Anatomical Labeling, Branch-point Matching, and Quantitative Analysis of Human Airway in Volumetric CT Images*. PhD thesis, University of Iowa, Iowa City, IA, August 2003.
12. P. K. Saha and J. K. Udupa, "Relative Fuzzy Connectedness among Multiple Objects: Theory, Algorithms, and Applications in Image Segmentation," *Computer Vision and Image Understanding* **82**, pp. 42–56, 2001.
13. K. Palágyi, E. Sorantin, E. Balogh, A. Kuba, C. Halmai, B. Erdohelyi, and K. Hausegger, "A Sequential 3D Thinning Algorithm and its Medical Applications," in *17th Int. Conf. Information Processing in Medical Imaging, IPMI 2001, Davis, CA, USA. Lecture Notes in Computer Science 2082*, pp. 409–415, 2001.
14. J. Tschirren, K. Palágyi, J. M. Reinhardt, E. A. Hoffman, and M. Sonka, "Segmentation, Skeletonization, and Branchpoint Matching — A Fully Automated Quantitative Evaluation of Human Intrathoracic Airway Trees," in *MICCAI 2002, Tokyo, Japan*, 9 2002.
15. M. Unser, A. Aldroudi, and M. Eden, "B-Spline Signal Processing: Part I – Theory," *IEEE Trans. Signal Processing* **41**, pp. 821–833, Feb. 1993.
16. M. Unser, A. Aldroudi, and M. Eden, "B-Spline Signal Processing: Part II – Efficient Design and Applications," *IEEE Trans. Signal Processing* **41**, pp. 834–848, Feb. 1993.
17. M. Sonka, G. K. Reddy, M. D. Winniford, and S. M. Collins, "Adaptive Approach to Accurate Analysis of Small-Diameter Vessels in Cineangiograms," *IEEE Trans. Med. Imag.* **16**, pp. 87–95, Feb. 1997.
18. Microsoft, *The Microsoft Developer Network (MSDN)*.
19. A. V. Goldberg and S. Rao, "Beyond the Flow Decomposition Barrier," *Journal of the ACM (JACM)* **45**, pp. 783–797, 1998.
20. Y. Shiloach and U. Vishkin, "An  $o(n^2 \log n)$  parallel max-flow algorithm," *Journal of Algorithms* **3**, pp. 128–146, 1982.
21. R. J. Anderson and J. C. Setubal, "On the Parallel Implementation of Goldberg's Maximum Flow Algorithm," in *Proceedings of the Fourth Annual ACM Symposium on Parallel Algorithms and Architectures*, pp. 168–177, (San Diego, California, United States), 1992.