Globally Optimal Segmentation of Interacting Surfaces with Geometric Constraints

Kang Li*, Xiaodong Wu†, Danny Z. Chen‡ and Milan Sonka*
*Department of Electrical and Computer Engineering
The University of Iowa, Iowa City, IA 52242–1595, Email: {kanli, sonka}@engineering.uiowa.edu
†Department of Computer Science
The University of Texas – Pan American, Edinburg, TX 78539, Email: xwu@cs.panam.edu
‡Department of Computer Science and Engineering
The University of Notre Dame, Notre Dame, IN 46556, Email: chen@cse.nd.edu

***Abstract*— Efficient detection of globally optimal surfaces representing object boundaries in volumetric datasets is important and remains challenging in many medical image analysis applications. We have developed an optimal surface detection method that is capable of simultaneously detecting multiple interacting surfaces, in which the optimality is controlled by the cost functions designed for individual surfaces and several geometric constraints defining the surface smoothness and interrelations. The method solves the surface detection problems by transforming them into computing minimum s-t cuts in the derived edge-weighted directed graphs. The proposed algorithm has low-order polynomial complexity and is computationally efficient. The method has been validated on over 100 computer generated volumetric images and 96 CT-scanned datasets of different-sized plexiglas tubes, yielding highly accurate results (mean signed error of the measured inner- and outer-diameters of the plexiglas tubes was $0.21 \pm 3.20\%$). Our approach can be readily extended to higher dimensional image segmentation.**

I. Introduction

The task of identifying globally optimal three-dimensional surfaces representing object boundaries is important in automated segmentation of volumetric medical images. Many computer methods have been developed for optimal segmentation of 2-D medical image data. 2-D edge-based segmentation utilizing graph-searching principles has become one of the best understood and most utilized medical image segmentation tools. However, previous attempts [1]–[3] for extending the graph-based segmentation methods to higher dimensions had either made them computationally intractable or traded their ability to achieve global optimums for efficiency.

Recently, Wu and Chen introduced a method for d-D ($d \geq 3$) optimal hypersurface detection that made true optimal surface segmentation in volumetric images practical [4], [5]. By modeling the problem as a directed geometric hypergraph, the method transformed the segmentation problem into computing the minimum s-t cut of a graph, which simplifies the problem and consequently solves it in polynomial time.

While the single optimal surface detection can be modeled by a 3-D geometric hypergraph [4], our novel method attempts to approach the simultaneous detection of k ($k \geq 2$) interacting surfaces by modeling the problem in a 4-D geometric hypergraph, where the fourth dimension consists of a set of special edges that model the interrelations between pairs of the desired surfaces. We show that the apparently daunting consequence of combinatorial explosion can be avoided by transforming the problems into computing minimum s-t cuts in similar ways as in [4]. As an extension to the original single surface detection algorithm, the new approach not only guarantees the optimality quality of the output surfaces based on the cost functions used, but is flexible and computationally efficient as well.

II. Problem Modeling

A. Single Surface Detection

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

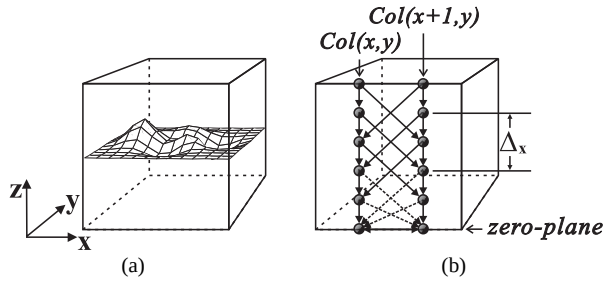


Fig. 1. The single surface detection problem. (a) The surface orientation. (b) Two adjacent columns of the constructed directed geometric hypergraph. Edges shown in dashed lines are optional.

to the vertex $V(x, y, z-1)$, denoted by $(V(x, y, z), V(x, y, z-1))$.

The smoothness constraints along the $\mathbf{x}$- and $\mathbf{y}$-directions are enforced in the following way. WLOG, the 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]–[3]). 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.

B. Multiple Surface Detection

In simultaneously detecting k ($k \geq 2$) distinct but interrelated surfaces, the optimality is not only determined by the inherent costs and smoothness properties of the individual surfaces, but also confined by their interrelations.

If surface interactions are not considered, the k surfaces S_i can be detected in k separate 3-D digraphs $G_i = (V_i, E_i)$ ($i = 1, 2, \ldots, k$), each of which is constructed in the way presented above. The vertex costs are computed utilizing k cost functions (not necessarily distinct), each one of which is used for the search of one surface. Taking the surface interrelations into account, another edge set E_s needs to be constructed modeling these interrelations, forming a geometric digraph $G(V, E)$ in 4-D space with $V = \bigcup_{i=1}^{k} V_i$ and $E = \bigcup_{i=1}^{k} E_i \cup E_s$. Constructions of E_s and V_s are detailed below.

Note that the surface interactions are specified by their pairwise relations. For each pair of the surfaces, our approach defines their relations using two parameters, $\delta^l \geq 0$ and $\delta^u \geq 0$, representing the surface *separation constraints*. The ideas are illustrated by examples as follows.

In many practical problems, the surfaces are expected not to intersect or overlap. For instance, the inner and outer tissue walls should be non-crossing, and the distance between them should be within some expected range in medical images. Suppose that for the two surfaces S_1 and S_2 being detected, the prior knowledge requires S_2 being below S_1. Let the minimum distance δ^l between them be 2 voxel units, and the maximum distance δ^u be 5 voxel units. Let the 3-D graphs used for the search of S_1 and S_2 be G_1 and G_2, respectively, and let $Col_1(x, y)$ and $Col_2(x, y)$ denote two corresponding columns in G_1 and G_2. For vertex $V_1(x, y, z)$ in $Col_1(x, y)$ with $z > \delta^u$, a directed edge in E_s connecting $V_1(x, y, z)$ to $V_2(x, y, z-\delta^u)$ is constructed. While for each vertex $V_2(x, y, z)$ with $z < Z - \delta^l$, a directed edge in E_s connecting $V_2(x, y, z)$ to $V_1(x, y, z+\delta^l)$ is introduced. This construction is applied to every pair of corresponding columns of G_1 and G_2. Due to the separation constraints, i.e., S_2 is at least δ^l voxel units below S_1, each vertex $V_1(x, y, z)$ with $z < \delta^l$ cannot be on surface S_1. Otherwise, no vertex in $Col_2(x, y)$ could be on surface S_2. Likewise, each vertex $V_2(x, y, z)$ with $z \geq Z - \delta^l$ cannot belong to surface S_2. Hence, for each column $Col_1(x, y) \in G_1$, it is safe to remove all vertices $V_1(x, y, z)$ with $z < \delta^l$ and their incident edges in E_1. Meanwhile, we need to reassign a cost $c(x, y, \delta^l)$ of voxel $\mathcal{I}(x, y, \delta^l)$ to vertex $V_1(x, y, \delta^l)$ and add a directed edge in E_s connecting $V_1(x, y, \delta^l)$ to $V_2(x, y, 0)$. Note then the *zero-plane* (i.e., the set of vertices with $z = 0$) of G_1 consists of all vertices $V_1(x, y, z)$ with $z = \delta^l$, which forms a feasible surface in G_1. To ensure that, a set of directed edges in E_s is introduced to make the zero-plane of G_1 strongly connected. While for each column $Col_2(x, y) \in G_2$, we safely eliminate all vertices $V_2(x, y, z)$ with $z \geq Z - \delta^l$ and their incident edges in E_2.

In other situations, we may allow the two interacting surfaces to cross each other. For these problems, instead of modeling the minimum and maximum distances between them, δ^l and δ^u specify the maximum distances that a surface can vary below and above the other surface, respectively. Incorporating this case in our example, every vertex $V_1(x, y, z)$ in G_1 shall connect to $V_2(x, y, max(0, z-\delta^l))$ in G_2; and every vertex $V_2(x, y, z)$ in G_2 has an edge to $V_1(x, y, max(0, z-\delta^u))$ in G_1. A summary of all these cases is presented in Figure 2.

III. The Algorithm

The multiple surface detection problem is formulated as computing a minimum s-t cut in a graph constructed from $\mathcal{I}$. Our approach is inspired by Wu and Chen's algorithms for the optimal net surface problems [4]. However, instead of searching for one surface, our algorithm concurrently identifies k surfaces, which achieve optimality in a more general setting. Furthermore, the time bound of our algorithm is independent of both the smoothness parameters (i.e., $\Delta_{\mathbf{x}}$ and $\Delta_{\mathbf{y}}$) and the surface separation parameters (i.e., ${\delta^l}_{i,i+1}$ and ${\delta^u}_{i,i+1}$, with

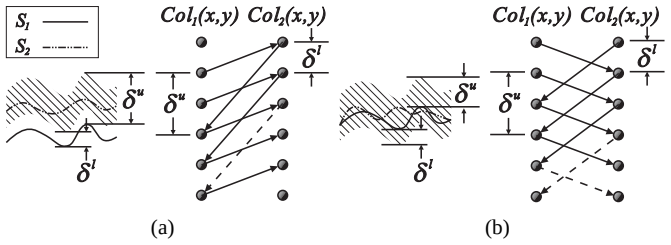


Fig. 2. Summary of surface interrelation modeling. S_1 and S_2 are two desired surfaces. $Col_1(x, y)$ and $Col_2(x, y)$ are two corresponding columns in the constructed graphs. Edges shown in dashed lines are optional. (a) The non-crossing case. (b) The case with crossing allowed.

$i \in \{1, 2, \ldots k-1\}$). Note that, the multiple surface detection problem could be infeasible, i.e., no k surfaces satisfying the geometric constraints exist in $\mathcal{I}$. The feasibility of the problem is easy to determine. Thus, we assume in this section the multiple-surface detection problem is feasible.

A. The Minimum Closed Set

In Section II, the construction of a weighted directed graph $G = (V, E)$ from the volumetric dataset $\mathcal{I}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ was described. The graph G consists of k disjoint 3-D sub-graphs $\{G_i = (V_i, E_i) : i = 1, 2, \ldots, k\}$, each of which is dedicated to searching for one surface. The separation constraints between any two surfaces are enforced in G by edges between the corresponding sub-graphs. Actually, the goal of such a construction is to guarantee that:

1) Any feasible k surfaces 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 feasible k surfaces in $\mathcal{I}$ with the same total cost.

A *closed set* $\mathcal{C}$ in a directed graph is a set of vertices such that all successors of any vertex in $\mathcal{C}$ are also contained in $\mathcal{C}$ [7], [8]. The *cost* of a closed set $\mathcal{C}$, denoted by $w(\mathcal{C})$, is the total cost of vertices in $\mathcal{C}$. Note that the closed set $\mathcal{C}$ in a graph can be empty (with a cost zero).

Recall the graph constructed in Section II. Note that in any G_i, if $V_i(x, y, z)$ is in a closed set $\mathcal{C}$ of G, then all vertices "below" it on $Col_i(x, y)$ are in $\mathcal{C}$. We denote the set of these vertices by $B_i^z(x, y)$, i.e., $B_i^z(x, y) = \{V_i(x, y, z') : z' \leq z\}$. The cost of $B_i^z(x, y)$ equals the cost of voxel $\mathcal{I}(x, y, z)$. In general, we are able to prove the following lemma, showing that computing the optimal k surfaces in $\mathcal{I}$ is equivalent to finding a non-empty closed set $\mathcal{C}^*$ of the minimum cost in G (the proof is omitted due to the space limitation).

Lemma 1: A non-empty closed set $\mathcal{C}^*$ of the minimum cost in G specifies the optimal k surfaces 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 for defining the optimal k surfaces in $\mathcal{I}$. This problem can be easily solved by performing a *translation* operation on G as introduced in [4]. 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) and Figure 2) are optional. The vertices of G_i 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 a very interesting observation: the graph is actually getting simpler (i.e., with less edges) as the smoothness constraints are relaxed (i.e., $\Delta_{\mathbf{x}}$ and/or $\Delta_{\mathbf{y}}$ become larger). This behavior is just the opposite to the traditional graph-search based algorithms for the problem.

B. Computing the Optimal k Surfaces

Based on Lemma 1, we need to compute a minimum-cost *non-empty* closed set C^* in G, which is a well studied problem in graph theory. As in [4], [7], [8], we compute C^* in G by transforming it into computing a minimum s-t cut in a *related* graph G_{st}. Note that the graph G_{st} has $O(kn)$ vertices and $O(kn)$ edges. Therefore, 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.

The optimal k surfaces based on $\mathcal{C}^*$ can be recovered in the following way. For each i ($i = 1, 2, \ldots, k$), recall that the sub-graph G_i is used to search for the target surface S_i. For every $0 \leq x < X$ and $0 \leq y < Y$, let $B_i(x, y) = \mathcal{C}^* \cap Col_i(x, y)$. Denote by $V_i(x, y, z^*)$ the vertex in $B_i(x, y)$ with the largest $\mathbf{z}$-coordinate. Then, voxel $\mathcal{I}(x, y, z^*)$ is on the i-th optimal surface S_i^*. In this way, the minimum closed set $\mathcal{C}^*$ of G defines the optimal k surfaces $\{S_1^*, S_2^*, \ldots, S_k^*\}$ in $\mathcal{I}$.

To sum up, we have the following theorem.

Theorem 1: The optimal k surfaces in a 3-D image $\mathcal{I}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ with n voxels can be computed in $T(kn, kn)$ time.

Finally, the outline of the algorithm is:

- Inputs: k, $\Delta_{\mathbf{x}}$, $\Delta_{\mathbf{y}}$, δ^l, δ^u and the cost function(s).
- Construct G ($= \bigcup_{i=1}^{k} G_i$) according to Section II.
- Transform G into G_{st} as mentioned above.
- Compute the minimum s-t cut of G_{st}.
- Recover the k optimal surfaces.

IV. Experiments

A. Data

To validate the correctness of the modeling techniques, we tested our method on a set of computer-generated phantoms of various sizes and surface positioning. Computer phantoms contained two or more surfaces with variable shapes and mutual positions, and the surfaces did not intersect. The datasets were blurred and superimposed with slight Gaussian noise to make the problem more realistic. Figures 3 and 4 provide examples of the utilized phantoms. Phantom sizes ranged from $16 \times 16 \times 16$ to $256 \times 256 \times 256$ voxels.

To design a surface detection cost function that would be appropriate for CT image data, a physical phantom was imaged by multi-detector CT and analyzed using our surface detection method. The phantom contains six plexiglas tubes,

numbered 1 through 6, with nominal inner diameters of 1.98, 3.25, 6.40, 6.50, 9.50 and 19.25 mm, respectively. The corresponding outer diameters are 4.45, 6.30, 9.70, 12.60, 15.60 and 25.50 mm, respectively. The phantom was scanned using Philips Mx8000 4-slice CT scanner with 3 different scan settings (*low dose*, *regular dose*, and *high dose*). Under each setting, the scans were taken at the 4 distinct angles of $0°$, $5°$, $30°$, and $90°$, rotated in the coronal plane, resulting in a total of 12 datasets for use in the validation. The regular dose scanning was intentionally repeated, yielding another 4 datasets used for initial calibration of the cost functions. In all cases, a resolution of $0.39 \times 0.39 \times 0.6$ mm^3 was used, images consisted of 200-250 slices, 512×512 pixels each.

B. The Cost Functions

Designing a cost function is of paramount importance for any graph based segmentation method. Simple cost functions reflecting brightness of the sought surfaces were used in the computer phantoms. In CT images, the cost function needs to be edge based and needs to incorporate imaging properties of the CT scanner as well as image properties of the surfaces. Therefore, the cost functions used for CT image segmentation utilized a combination of first and second derivatives of 2-D gray-level images [9]. We designed an approach in which plexiglas phantom tubes with known sizes were utilized for cost function design as well as for segmentation accuracy assessment. In the design stage, the known inner and outer diameters of 6 phantom tubes in 4 data sets (24 phantom tubes total) served as independent standard for constructing the cost functions used for CT images. Using the cost functions for the inner and outer borders, a different set of 12 scans of the same phantom (72 tubes) was used for accuracy validation of our method under different CT scanning conditions. The same cost functions were employed for detection of inner and outer airway wall surfaces in the pulmonary CT images.

C. Performance Indices

The performance of the method was tested in two ways. First, running times were recorded and compared in 50 computer-generated phantoms of varying sizes to gain a basic understanding of the speed/size relationships. All experiments were conducted on an AMD Athlon MP 2000+ (1.67GHz) Dual CPU workstation with 3.5 GB of memory running Microsoft Windows XP. The running-times were measured three times and the results were averaged.

Second, surface detection accuracy was determined in physical phantoms in comparison with the independent standard. The comparison was conducted by measuring the minor and major diameters, d_{ma} and d_{mi}, of the segmented surfaces in the middle $1/3$ of tube (about 40 slices), then computing the mean $\pm$ standard deviation of the percentage signed difference between nominal diameters and average measured diameters $\bar{d} = (d_{ma} + d_{mi})/2$.

D. Segmentation of Wall Surfaces of Plexiglas Tubes

The phantom tubes and in vivo airway walls were segmented using the cost functions described above. The phantom and airway tree segmentation procedures are identical in principle. The segmentation process consisted of the following steps: 1) topological pre-segmentation, yielding approximate spatial locations of the tubes; 2) tube skeletonization; 3) 3-D image resampling following centerlines of the tubes; 4) tube wall segmentation (using the proposed algorithm), and 5) quantitative measurement.

V. Results

A. Computer Phantoms

Many computer phantoms were analyzed in the early stages of the algorithm development. As an example, results are presented for phantoms with several surfaces embedded in a volumetric image. The surface cross-sections are shown in Figure 3(a). The image was generated so that the surfaces consist of voxels of lower gray values compared to the background. As shown, three separate surfaces are embedded in the image. The lower two surfaces are smoother, while the topmost surface is rough. When simultaneously searching for 2 out of the 3 surfaces in the original image, the lowest surface was fixed by setting the first cost-function to have low magnitude only at the position of the lowest surface (Figure 3(a); the surface locking details are given in Section VI-A). The second cost function yielded an identical magnitude at all graph nodes of the middle and topmost surface positions (Figure 3(c)). In this case, the resulting surface detection is fully controlled by the smoothness constraints.

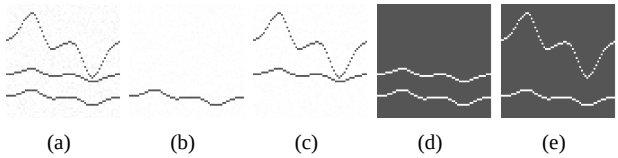


Fig. 3. Effect of intra-surface smoothness constraints. (a) Cross-section of the original image. (b) The first cost image. (c) The second cost image. (d) Results obtained with smoothness parameters $\Delta_{\mathbf{x}} = \Delta_{\mathbf{y}} = 1$. (e) Results obtained with smoothness parameters $\Delta_{\mathbf{x}} = \Delta_{\mathbf{y}} = 5$.

The results are shown in Figures 3(d) and (e). In the first case, the smoothness parameters $\Delta_{\mathbf{x}}$ and $\Delta_{\mathbf{y}}$ were both set to 1. The topmost surface apparently does not satisfy these criteria and, as expected, the algorithm detected the middle surface. In the second case, the smoothness parameters were set so that both the middle and topmost surfaces satisfied the constraints and had the same cost values. When this happens, the algorithm will identify the surface closest to the zero-plane (i.e., with $z = 0$). In this case, since the coordinate origin is at the top-left corner, the resulting surface will be the topmost surface.

Figure 4 shows the result of a triple-surface detection experiment. The surfaces in the dataset are 10 voxels apart. The algorithm was set to always identify the lowest surface and select 2 out of the 3 surfaces above it, which was fully controlled by the separation constraints.

(a) (b) (c) (d) (e)

Fig. 4. Effect of surface separation constraints. (a) Cross-section of the original image. (b) The first cost image. (c) The second/third cost image. (d) Results obtained with separation parameters $\delta_{12}^{l} = 5$, $\delta_{12}^{u} = 15$ and $\delta_{13}^{l} = 16$, $\delta_{13}^{u} = 25$. (d) Results obtained with separation parameters $\delta_{12}^{l} = 5$, $\delta_{12}^{u} = 15$ and $\delta_{13}^{l} = 26$, $\delta_{13}^{u} = 35$.

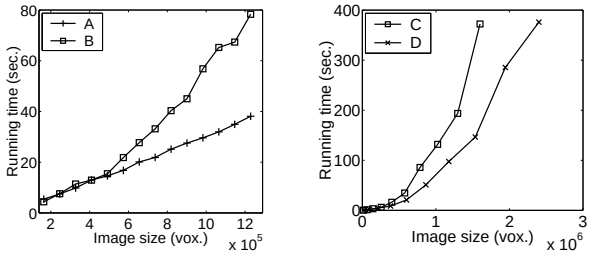
B. Execution Times

The average execution times of our simultaneous k-surface ($k = 2, 3$) detection algorithm on datasets of different sizes are shown in Table I.

TABLE I
AVERAGE EXECUTION TIMES

Image Size	k = 2	k = 3	Image Size	k = 2	k = 3
$40 \times 40 \times 40$	1.2	1.7	$100 \times 100 \times 40$	16.2	22.3
$60 \times 60 \times 40$	3.3	4.8	$140 \times 140 \times 40$	85.8	96.1
$80 \times 80 \times 40$	6.1	9.4	$200 \times 200 \times 40$	376.1	401.3

As shown in Figure 5(a), the speed of our method relates differently to the width and height of the image. Assuming that the surface is oriented as shown in Figure 1, the width refers to the X- or Y-size of the image, and the height refers to the Z-size. The results indicate that the execution time has an approximately linear relationship with the image height, whereas its relationship with the image width is captured by a low-order polynomial. The accompanying Figure 5(b) reveals the same fact from a different angle.

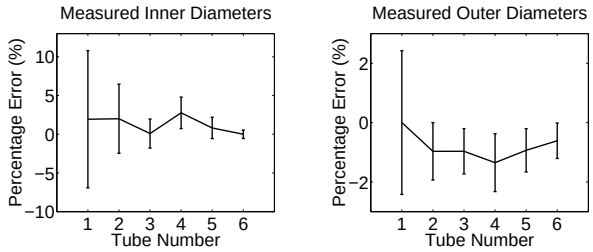


(a) (b)

Fig. 5. Average execution times. (a) Triple-surface detection case. Curve A corresponds to the testing sets of size $64 \times N \times 64$, and curves B is for the testing sets of size $64 \times 64 \times N$, where N varies from 40 to 300. (b) Double-surface detection case. Testing sets are of sizes $M \times M \times 40$ and $M \times M \times 60$, where M varies from 20 to 200.

C. Accuracy Assessment in Physical Phantom Tubes

Errors of the computer segmented and measured diameters are presented in Figure 6. The overall signed errors for the inner and outer borders are $1.27 \pm 4.08\%$ and $-0.85 \pm 1.23\%$, respectively. In general, the cost functions, which were trained on standard-dose CT images of phantoms, tend to overestimate the inner diameters and underestimate the outer ones in high-dose phantom CT images although the errors stay small.



(a) (b)

Fig. 6. Signed errors of inner- and outer-diameter measurements of the plexiglas tubes. Horizontal axes represent the nominal diameters. Vertical axes represent the signed percent error. Mean error $\pm$ SD is shown for each sample.

VI. DISCUSSION

A. Surface Locking

In the computer phantom experiments reported in Section V-A, we mentioned that the algorithm was set to fix the position of one surface. We refer to this behavior descriptively as *surface locking*, which was achieved in those experiments by manipulating the cost functions. Actually, with our algorithm, there is another way to lock the positions of one or more resulting surfaces or, more generally, force an arbitrarily specified set of voxels to be part of the final result. This is achieved by changing the graph construction itself: we can make the (x, y)-columns containing the corresponding locked vertices contain only those vertices, i.e., each of these columns has only one vertex, which corresponds to one locked voxel. Certainly, the smoothness constraints and surface separation constraints must be satisfied.

B. Variable Geometric Constraints

In this paper, we have only considered homogeneous smoothness and separation constraints, whose values remain constant along each graph dimension. In fact, the proposed algorithm allows variable geometric constraints as well, which can be specified adaptively based on different image context. This flexibility has certain practical implications. In some problems, for instance, the surface smoothness and interrelations may vary at different locations. The variable geometric constraints can be incorporated into the algorithm by rearrangements of the graph edges. As a result, the edges will no longer be parallel as they are in the constant-constraint case. For the variable-constraint setup to become practical, a key problem must be solved and still remains challenging: how to automatically adjust the geometric constraints as needed. Some machine learning approaches may be applied, which will be a future research topic.

C. Advantages and Limitations

The algorithm efficiently detects the globally optimal surfaces in the entire region-of-interest (ROI), enabling highly accurate image segmentation that is an prerequisite of reliable quantitative image analyses. Its ability to encode the interrelations of any number of surfaces is unprecedented.

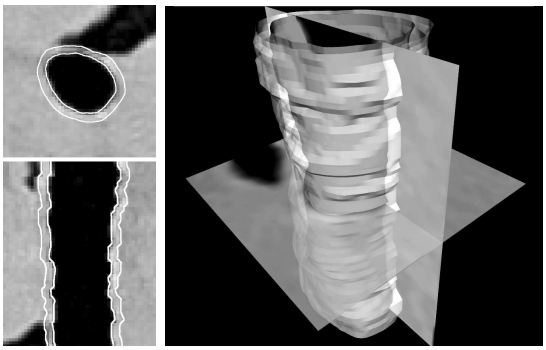
Unlike the level set driven techniques [10], [11], the algorithm cannot handle topology changes. An example of such a limitation is the inability to directly segment branching tubular structures. However, as pointed out in [12], topology flexibility is not always desired. Another apparent limitation 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 subjects of our algorithm since there is no perfectly invertible method to unfold them.

D. Implementation and Improvements

Since the segmentation problem has been transformed into solving a minimum s-t cut in directed geometric hypergraphs, the performance of the approach crucially hinges on the minimum s-t cut algorithm being used. Our implementation used the minimum s-t cut algorithm reported in [13]. For other implementation issues, we refer the reader to our earlier work [5]. To enable real-time application, further improvements can be made by adopting more sophisticated min s-t cut/max flow algorithms (for example, see [14]). Another option is to parallelize the algorithm. The multi-scale approach commonly used in image processing provides yet another alternative to speed up our method.

E. Application to In Vivo Medical Data

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 segmentation [15], and applied to concurrently segmenting the inner and outer wall surfaces of intrathoracic airways imaged by multi-detector CT. The time for segmenting an entire airway-tree (32 cylindrical segments) in a $512 \times 512 \times 550$ dataset was approximately 6 minutes. Preliminary results revealed higher accuracy and 3-D consistency compared to the traditional 2-D dynamic programming methods [5].



(a) (b)

Fig. 7. Segmented inner and outer walls of human pulmonary airways imaged with multi-detector CT. (a) The transversal and sagittal cross-sections of an airway segment. (b) 3-D view of the analyzed airway segment.

VII. Conclusion

A polynomial-time algorithm for simultaneously detecting multiple mutually related surfaces in volumetric images has been developed and validated in computer generated and physical phantom images. The method is efficient and robust. The resulting surfaces are globally optimal with respect to the employed objective functions. The surface smoothness and separation parameters provide a flexible means for modeling various inherent properties and interrelations of the desired surfaces. The method is readily extensible to higher dimensions.

Acknowledgment

The authors thank Dr. Juerg Tschirren for providing experimental data and algorithm testing environment. CT imaging was performed under guidance of Drs. Eric A. Hoffman and Geoffrey McLennan. The research was supported, in part, by NIH NHLBI grants R01-HL64368, R01-HL63373, R01-HL071809, NSF Grant CCR-9988468 and a grant from the Computing and Information Technology Center at the University of Texas – Pan American, Edinburg, Texas, USA.

References

[1] D. Thedens, D. Skorton, and S. 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*, Sept. 1990, pp. 57 –60.

[2] D. R. Thedens, D. J. Skorton, and S. R. Fleagle, "Methods of graph searching for border detection in image sequences with applications to cardiac magnetic resonance imaging," *IEEE Trans. Med. Imag.*, vol. 14, no. 1, pp. 42–55, 1995.

[3] R. J. Frank, "Optimal surface detection using multi-dimensional graph search: Applications to Intravascular Ultrasound," Master's thesis, The University of Iowa, 1996.

[4] 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)*, Malaga, Spain, July 2002, pp. 1029–1042.

[5] K. Li, X. Wu, D. Z. Chen, and M. Sonka, "Efficient optimal surface detection: Theory, implementation and experimental validation," in *Proc. SPIE International Symposium on Medical Imaging*, Feb. 2004, in press.

[6] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 2nd ed. Brooks/Cole Publishing Company, 1999.

[7] J. C. Picard, "Maximal closure of a graph and applications to combinatorial problems," *Management Science*, vol. 22, pp. 1268–1272, 1976.

[8] D. S. Hochbaum, "A new-old algorithm for minimum-cut and maximum-flow in closure graphs," *Networks*, vol. 37, pp. 171–193, 2001.

[9] M. Sonka, G. K. Reddy, M. D. Winniford, and S. M. Collins, "Adaptive approach to accurate analysis of small-diameter vessels in cineangiograms," pp. 87–95, Feb. 1997.

[10] J. A. Sethian, *Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, 2nd ed. Cambridge, U.K.: Cambridge University Press, 1999, 400 pages.

[11] S. J. Osher and R. P. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, 1st ed. Springer Verlag, 2002, 296 pages.

[12] X. Han., C. Xu, and J. L. Prince, "A topology preserving level set method for geometric deformable models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, pp. 755–768, June 2003.

[13] 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.

[14] A. V. Goldberg and S. Rao, "Beyond the flow decomposition barrier," *Journal of the ACM (JACM)*, vol. 45, pp. 783–797, 1998.

[15] J. Tschirren, "Segmentation, anatomical labeling, branch-point matching, and quantitative analysis of human airway in volumetric CT images," Ph.D. dissertation, University of Iowa, Iowa City, IA, August 2003.