



ELSEVIER

Computers in Biology and Medicine 36 (2006) 974–996

Computers in Biology
and Medicine

www.intl.elsevierhealth.com/journals/cobm

Quantitative analysis of pulmonary airway tree structures

Kálmán Palágyi^{a,1}, Juerg Tschirren^a, Eric A. Hoffman^{b,c}, Milan Sonka^{a,*}

^a*Department of Electrical and Computer Engineering, The University of Iowa, Iowa City, IA 52242, USA*

^b*Department of Biomedical Engineering, The University of Iowa, Iowa City, IA 52242, USA*

^c*Department of Radiology, The University of Iowa, Iowa City, IA 52242, USA*

Received 15 October 2004; accepted 11 May 2005

Abstract

A method for computationally efficient skeletonization of three-dimensional tubular structures is reported. The method is specifically targeting skeletonization of vascular and airway tree structures in medical images but it is general and applicable to many other skeletonization tasks. The developed approach builds on the following novel concepts and properties: fast curve-thinning algorithm to increase computational speed, endpoint re-checking to avoid generation of spurious side branches, depth-and-length sensitive pruning, and exact tree-branch partitioning allowing branch volume and surface measurements. The method was validated in computer and physical phantoms and in vivo CT scans of human lungs. The validation studies demonstrated sub-voxel accuracy of branch point positioning, insensitivity to changes of object orientation, and high reproducibility of derived quantitative indices of the tubular structures offering a significant improvement over previously reported methods ($p \ll 0.001$).

© 2005 Elsevier Ltd. All rights reserved.

Keywords: 3D skeletonization; Vascular tree; Airway tree; Computed tomography; Lung; Validation

1. Introduction

Tubular structures are frequently found in living organisms. The tubes—e.g., arteries or veins—are organized into more complex structures. Trees consisting of tubular segments form the arterial and venous systems, intrathoracic airways form bronchial trees, and other examples can be found. Com-

* Corresponding author. Tel.: +1 319 335 6052; fax: +1 319 335 6028.

E-mail address: milan-sonka@uiowa.edu (M. Sonka).

¹ K. Palágyi worked on the topic during his 1-year visit of the Departments of Electrical and Computer Engineering, The University of Iowa. He is now back with the University of Szeged, Hungary.

puted tomography (CT) or magnetic resonance (MR) imaging provides volumetric image data allowing identification of such tree structures. Frequently, the trees represented as contiguous sets of voxels must be quantitatively analyzed. The analysis may be substantially simplified if the voxel-level tree is represented in a formal tree structure consisting of a set of nodes and connecting arcs. To build such formal trees, the voxel-level tree object must be transformed into a set of interconnected single-voxel centerlines representing individual tree branches. Therefore, the aim of our work was to develop a robust method for identification of centerlines and bifurcation (trifurcation, etc.) points in segmented tubular tree structures acquired in vivo from humans and animals using volumetric CT or MR scanning, rotational angiography, or other volumetric imaging means.

There are many reasons why identifying tree skeletons is important. Skeletons can serve as one-dimensional structures allowing guidance for orderly exploration of the entire tree; they can serve as viewpoint trajectory for navigation purposes in virtual bronchoscopy or angioscopy. To facilitate quantitative analysis of the vascular or bronchial tree, e.g., luminal area or wall thickness, measurements must be obtained in cross-sections perpendicular to the long axis of the tree segments. Clearly, planes normal to the tree skeletons must be identified and skeleton correctness is of paramount importance. As such, quantitative assessment of asthma or cystic fibrosis from pulmonary CT images depends on the performance of the tree skeletonization method. Similarly, accuracy and reproducibility of arterial plaque thickness measurements from coronary CT or intravascular ultrasound depends on the ability to skeletonize tubular structures.

Many researchers focused on this task in the past. Gerig et al. [1] proposed a method for extracting 3D curvilinear structures from volume image data and transforming them into a symbolic description which represents topological and geometrical features of tree-like, filamentous objects. They applied the modified version of an early 3D parallel curve-thinning algorithm [2] in order to ensure the topological correctness. The thinned structures were transformed into a graph description. An estimation of the local width/diameter was obtained by distance transform [3] applied to the elongated segmented tree. The method was applied to the characterization of the cerebral vascular system segmented from MR angiography.

Wood et al. [4] suggested a method to reduce 3D lung tree structures (i.e., airways and pulmonary vessels) to their centerlines to facilitate measurement of branch segment length and angle. Their centerline extraction is a 2-step process: first, an initial structure is derived from the region-growing algorithm by assigning an integer distance descriptor to each iteration, then the centroid is calculated for each group of voxels with the same integer descriptor and the neighboring centroids are connected; finally, branch-points are repositioned by line-fitting. Branch length, branch angle, and cross-sectional area are measured as linear distance between two successive branch-points, the angle between two child-branches, and by area measurement in a 2D plane perpendicular to the centerline at a point, respectively. The method was validated in two Plexiglas phantom and an intact excised dog lung.

Mori et al. [5] presented a rendering method using tree structure of objects in virtualized bronchus endoscope system. The surface data which belongs to each branch is a set of triangle patches. The calculation of that set was based on the centerlines of the bronchus. Centerlines of the bronchus were extracted by applying a 3D curve-thinning algorithm [6], which was augmented with the Euclidean distance transformation.

Nyström [7] proposed a method to extract centerlines from 3D tree structures. She applied a 2-step process: first, surface-skeleton was extracted by an iterative distance-based method, then the surface-skeleton was reduced to a curve-skeleton (centerlines). Nyström used a simple iterative prun-

ing method for correcting the “hairy” centerlines: voxels labeled by “small” distances are deleted without disconnecting the centerlines. Note that significant branches are shortened and unwanted branches in “thick” segments are preserved. The complex method was tested in a synthetic tree and in arteries of the pelvis segmented from a magnetic resonance angiography scan.

Wan et al. [8] developed a method for extracting hepatic vasculature in rats using micro-CT images. They applied a 3D parallel curve-thinning algorithm [9] to determine the centerlines of the elongated segmented tree. Pruning is performed via a symbolic graph representation of the centerlines; a side branch is deleted if it is shorter than a pre-specified minimum length. Numerical comparison between known (manually determined) and automatically identified branch-points were performed for a phantom and three vascular trees. The mean coordinate-difference (positioning error) was 1 to 3.5 voxels. Application of this method to quantitative analysis of vascular trees was reported in [10].

Krissian et al. [11] presented a method for model-based detection of tubular structures in 3D images. They proposed a general differential geometry based approach for extracting centerlines and reconstructing the whole structure.

Mori et al. [12] suggested a method for automated anatomical labeling of the bronchial branches. Centerlines of the bronchus were extracted by applying a 3D curve-thinning algorithm [6], which was augmented with the Euclidean distance transformation. The graph representation was derived from the thinned tree. They applied the a distance-based 3D curve-thinning algorithm [6]. A side branch was eliminated if its length was shorter than a pre-specified minimum length, or the corresponding branch-point was not closer to the surface than a given threshold. The method was validated in 14 CT-scanned studies.

Toriwaki and Mori [13] proposed a new distance-based sequential 3D curve-thinning algorithm for extracting centerlines from 3D tubular structures. They suggested a sophisticated parameter-controlled pruning method using length and distance information. Their method was applied by Sera et al. [14] for analyzing the morphometry of small airways from microfocal X-ray computed tomography.

Aylward and Bullitt [15] presented a differential geometry based approach for centerline extraction from tubular objects. Their method is based on intensity ridge traversal.

Maddah et al. [16] developed a 5-step method for extracting centerlines. It consisted of the following steps: distance transform, automatic selection of endpoints, path initialization (i.e., initial paths are generated on the surface), iterative path centering, and path thinning (i.e., converting the two-voxel wide segments into one-voxel wide ones).

Chen and Molloy [17] proposed a method for automatic 3D vascular tree construction. They applied a parallel 3D curve-thinning algorithm [18] and a sophisticated pruning and tree construction algorithm. Their pruning algorithm used only length information which may cause limitations in real-world images.

Deschamps and Cohen [19] developed a method for fast extraction of tubular structure centerlines that may serve as trajectories for virtual endoscopy. Working with a similar application in mind, Bouix et al. [20] presented an automated method for computing central paths through tubular structures based on medial surface algorithms and minimal outward flux.

Antiga et al. [21] presented a centerline extraction method based on solving Eikonal equation on Voronoi diagram. Their approach provides maximum inscribed sphere estimation and geometric characterization of the surface.

Despite the wealth of previous work, no perfect skeletonization technique exists to date. Our new approach that is presented below is attempting to overcome most of the existing problems. Since our work is driven by the need for quantitative analysis of intrathoracic airway trees from multidetector CT

images, we concentrated on developing a method serving this purpose. However, the resulting approach is widely applicable to a variety of medical image data.

2. Methods

The reported method allows to quantitatively analyze tubular tree structures. Assuming that an imperfectly segmented tree was obtained from volumetric data in the previous stages, the presented technique allows to obtain a single-voxel skeleton of the tree while overcoming many segmentation imperfections, yields formal tree representation, and performs quantitative analysis of individual tree segments on a tree-branch basis. The input of the proposed method is a 3D binary image representing a segmented voxel-level tree object. All main components of our method were specifically developed to deal with imaging artifacts typically present in volumetric medical image data. As such, the method consists of the following main steps:

- (1) airway segmentation,
- (2) correction of the segmented tree,
- (3) identification of the tree root,
- (4) extraction of the 3D centerline—skeletonization,
- (5) tree pruning,
- (6) centerline smoothing,
- (7) identification of branch-points,
- (8) generation of a formal tree structure,
- (9) tree partitioning, and
- (10) quantitative analysis.

These steps are now described in more detail.

2.1. Airway segmentation

The employed airway segmentation method is based on fuzzy connectivity [22,23]. After automatically identifying a seed point in the upper trachea, two regions—foreground and background—are grown simultaneously and are competing against each other during the segmentation process. To minimize occurrence of leaks into the surrounding lung parenchyma during the region-growing process, the root of a leak is identified once the leak is observed and the segmentation parameters are modified. To accomplish such behavior, a relatively small adaptive region of interest (ROI) is defined in an iterative fashion. The ROI follows the airway tree branches as they are segmented. The ROI has a cylindrical shape and adapts its geometrical dimensions, its orientation, and position to the predicted size, orientation, and position of the airway branch to be segmented. Using a cylindrically shaped ROI (versus the more common cubical ROI used in other 3D image segmentation tasks) has the advantage that the ROI better adapts to the target shape, which is close to cylindrical. This means less “useless” background voxels have to be analyzed and the computing time can be shortened. A similar approach was independently used by Mori et al. [24]. After the series of iteration steps, an airway tree is segmented. The segmentation algorithm was shown

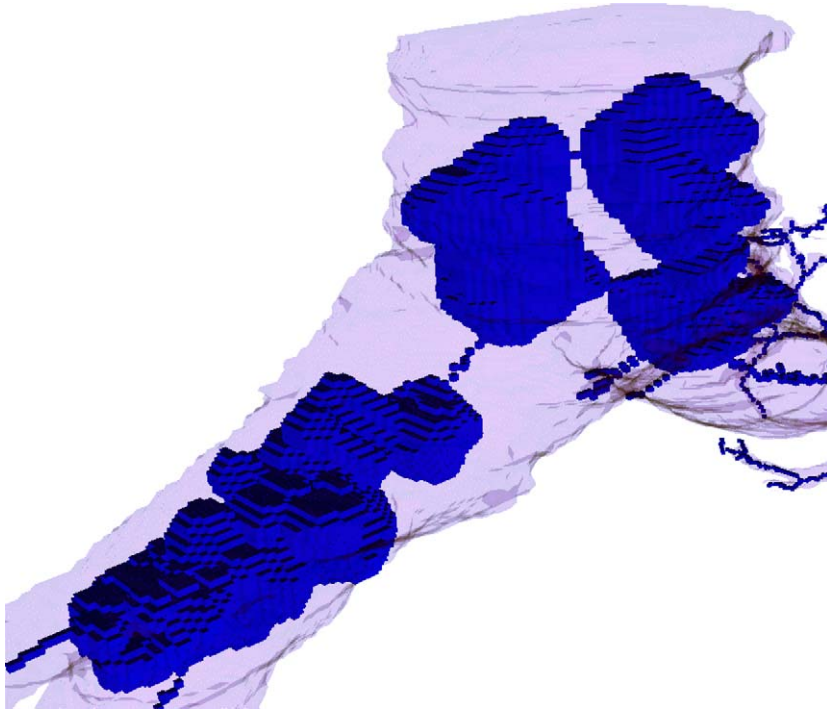


Fig. 1. Part of a segmented human airway tree containing some cavities. Its topological kernel shown superimposed (topological kernel is a minimal structure topologically equivalent to the original object). Each cavity of the structure corresponds to a cavity in the topological kernel thus appearing as an enlarged “bubble” with unit wide hull. Therefore, topologically correct skeletonization of any structure with a false cavity yields an incorrect formal representation since it cannot deal with cavity elimination.

robust in low dose and regular dose scans from normal and diseased subjects when compared to previous airway segmentation approaches [25].

2.2. Correction of the segmented tree

When applied to clinical volumetric images, segmentation algorithms may produce imperfect results in which the segmented objects contain internal cavities (i.e., connected “0” voxels surrounded by “1” voxels), holes (i.e., “0” voxels forming tunnels), and bays (i.e., disturbances without a topological change). Some of such imperfections cause unwanted changes of the underlying topology, and all of them disturb the centerline detection process and consequently yield an incorrect set of centerlines and thus incorrect formal tree representation (Fig. 1). To overcome the effects of artifactual cavities, the “0” voxels connected to the frame of the volume are first labeled by sequential forward and backward scanning (instead of the conventional object labeling). Then, all unlabeled “0” voxels are changed to “1” voxels. The applied method is similar to the linear-time Chamfer distance mapping [3]. As a result, all cavities are filled with no connectivity alteration.

Holes and bays are removed by applying morphological closing [26] (i.e., a dilation followed by an erosion with an experimentally determined structuring element). Note that the closing is a double-edged sword; it is suitable for filling small gaps, holes, and cavities, but new holes may be created in the process. This side effect can be handled by a post-processing pruning process. In this application, however, the object of interest is a single tubular structure and the potential negative effect of the closing operation is minimal.

2.3. Root detection

Our work deals with analysis of intrathoracic airway trees from volumetric CT (or MR) image data. Consequently, a priori knowledge of the data set is used to identify the tree root. In other applications, a different root identification approach may be needed. The root detection is not a critical phase of the process. It can be identified interactively or automatically [8,12].

In the pulmonary CT images, the center of the topmost nonzero 2D slice in direction z (detected by 2D shrinking) defines the root of the formal tree to be generated and belongs to the trachea. The detected root voxel acts as an anchor point during the centerline extraction (i.e., it cannot be deleted by the forthcoming iterative peeling process).

2.4. Centerline extraction

One of the well-known approaches to centerline determination is to construct a 3D skeleton of the analyzed object. However, some of the properties of 3D skeletons in discrete grids are undesirable. Specifically, in the case of 3D tubular objects, we do not need the exact skeleton, since a 3D skeleton generally contains surface patches (i.e., branched 2D manifolds) [27]. We need a skeletonization method that can suppress creation of such surface patches. As a solution, a 3D curve-thinning algorithm was developed that is preserving line-end points and can thus extract both geometrically and topologically correct centerlines. As part of this process, a method for endpoint re-checking was developed based on comparisons between the centerline configuration at some stage of thinning and the previous object configuration.

Thinning is a frequently used method for producing an approximation to the skeleton in a topology-preserving way [28]. Border points of a binary object that satisfy certain topological and geometric constraints are deleted in the iteration steps. In case of tubular 3D objects, thinning has a major advantage over other skeletonization methods. Curve-thinning (i.e., iterative object reduction preserving line-end points) can directly produce one voxel wide centerlines [29].

In order to outline our thinning scheme, let us first define $(26, 6)$ images, border points (corresponding to the six main directions in 3D), line-end points, and simple points. A binary image is $(26, 6)$ image if 26-connectivity (i.e., the reflexive and transitive closure of the 26-adjacency relation) is considered for “1” voxels forming the objects and 6-connectivity (i.e., the reflexive and transitive closure of the 6-adjacency) is considered for “0” voxels [28]. A “1” voxel in a $(26, 6)$ image is called U-border point if its 6-neighbor in direction U (“up”) is “0”. We can define N-, E-, S-, W-, and D-border (“down”) points in the same way. A “1” voxel is called line-end point if it has exactly one “1” 26-neighbor. A “1” voxel is called a *simple* point if its deletion does not alter the topology of the image [28]. It needs to be emphasized that simplicity in $(26, 6)$ images is a local property that can be decided by investigating the 26-neighbors, the $3 \times 3 \times 3$ neighborhood of any given point [30].

Our sequential thinning algorithm can be regarded as a modified version of the method proposed by Lee et al. [31]. Our algorithm is outlined as follows:

```

repeat
  for each direction U, N, E, S, W, and D do
    mark all border points according to the actual direction
    that are simple points and not line-end points
    for each marked point  $p$  do
      if  $p$  is simple in the actual image then
        if  $p$  is not a line-end point then
          delete  $p$ 
        else if  $\#(\text{deleted 6-neighbors of } p) \geq t$  then
          delete  $p$ 
      endifor
    endifor
  until no changes occur

```

One iteration step of the sequential object reduction process (i.e., the kernel of the **repeat** cycle) is decomposed into six successive sub-iterations according to the six main directions in 3D. Each sub-iteration consists of two phases; at first the border points of the actual type that are simple and not line-end points are marked as potential deletable points of the actual sub-iteration. This phase of the algorithm can be executed in parallel, but the forthcoming re-checking phase must be sequential. During the re-checking, a marked point is deleted if it remains simple and is not a line-end point after the deletion of some previously visited marked points. In addition, in the following special cases, simple points are also deleted if they have become line-end points. The algorithm uses an extra parameter $t \in \{0, 1, 2, 3, 4, 5, 6\}$ and a marked (simple and line-end) point can be deleted if at least t points of its 6-neighbors have been deleted during the actual phase of the process. This additional condition identifies the configurations which are somehow “convex” before deletion and become “sharp” after the deletion of some points since these configurations are likely to produce spurious branches. Note that if $t = 6$, the endpoint re-checking has no effect (since a point is not simple if all of its 6-neighbors belong to the object). In that case, it produces the same result as the method proposed by Lee et al. [31]. According to our experience, setting $t = 1$ or $t = 2$ is suggested for human airway trees. Fig. 2 demonstrates usefulness of the endpoint re-checking. Note that $t = 0$ may involve unwanted over-shrinkage (i.e., some important branches may be deleted), but it never causes over-shrinkage in a thinned (i.e., 1-voxel wide) structure.

2.5. Pruning

Unfortunately, each skeletonization algorithm (including ours) is rather sensitive to coarse object boundaries or surfaces. As a result, the produced (approximation to the) skeleton generally includes false segments that must be removed by a pruning step [32,33].

Applying a proper pruning method that would yield reliable centerlines is critical in all tree-skeletonization applications. An unwanted branch causes false generation numbering and consequently false measurements corresponding to the individual segments of the tree (including length, volume, surface area, etc.).

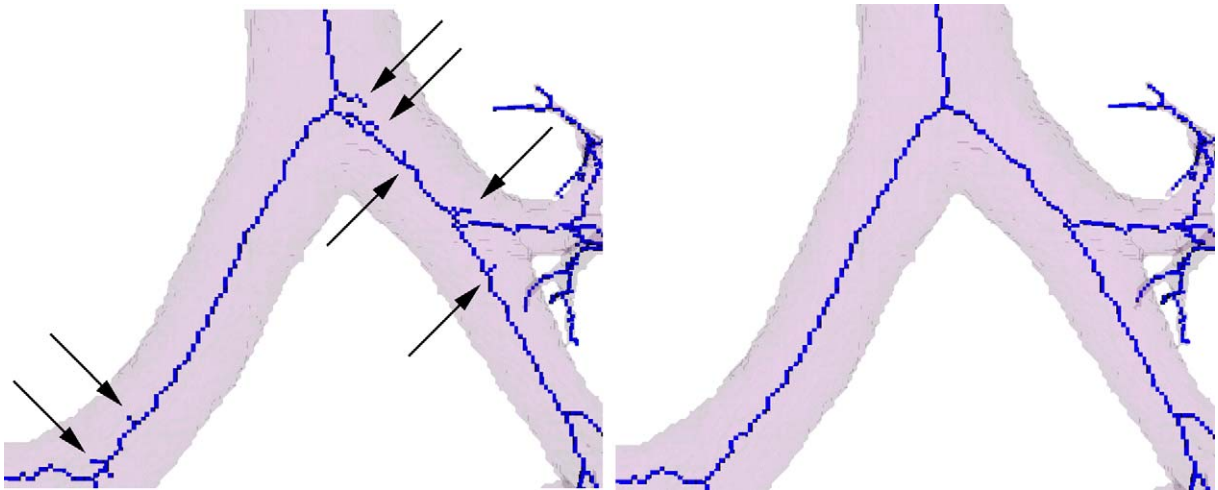


Fig. 2. A part of a segmented tree and its raw centerline extracted by the proposed algorithm without endpoint re-checking ($t = 6$) (left). The result with endpoint re-checking ($t = 1$) (right). The centerline extracted by endpoint re-checking contains 125 (true) branch-points (junctions) and 128 line-end points. There are 167 branch-points and 176 line-end points in the centerline generated without endpoint re-checking. Several of the unwanted branches are marked by arrows.

There are pruning approaches (e.g., morphological pruning [26]) capable of removing all side branches that are shorter than a predefined threshold [8]. Those methods necessarily fail in structures consisting of tubular segments of varying thickness. Therefore, we have developed a method capable of removing “long” parasitic branches from “thick” parts and preserving “short” correctly determined branches in “thin” segments. Spurious branches are identified by assessing the distance-from-surface function and the branch length [32]. Our pruning process consists of the following two phases:

- cutting holes that remain after the morphological closing, and
- deleting side branches using both the length and depth information.

At first, the centerlines are converted into a graph structure (each voxel corresponds to a graph node/vertex and there is an edge between two nodes if the corresponding voxels are 26-adjacent). Then, Dijkstra’s algorithm is applied to solve the single-source shortest-path problem [34] that maintains a rooted tree from a source node (i.e., the root detected in the first nonzero 2D slice in direction z). Since the result of Dijkstra’s algorithm is always a (cycle-free) tree, we can detect and cut holes (loops) in the centerlines easily: a skeletal point is to be deleted if it is not a line-end point and is not the parent of any other point in the Dijkstra’s tree (Fig. 3). This heuristic hole-cutting approach works well, although counter-examples can be given in which the heuristic does not apply (Fig. 3c).

After the hole cutting, the parasitic side branches shall be removed. We have developed a centerline pruning that uses both the branch length and the distance-from-surface (depth) information for the identification of a pruning candidate (Fig. 4). We delete all branches if their lengths are shorter than a given threshold t_l and their branch-points are not closer to the border/surface of the elongated tree (after topological correction) than a given threshold t_d . Note, that a similar but not identical pruning rule has been reported by Mori et al. [12]. They delete a branch if its length is shorter than a given threshold t_l or it

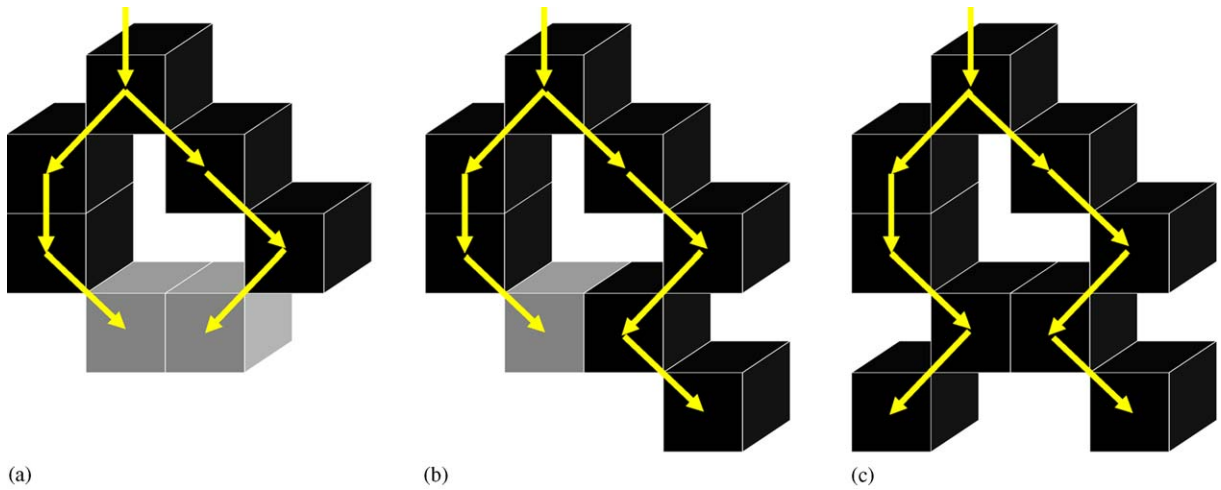


Fig. 3. Examples of the hole-cutting method. Dark voxels are preserved, bright ones are deleted. Arrows correspond to the edges in the Dijkstra's tree. (a,b) Holes are successfully eliminated in most cases, (c) but counter-examples can be found.

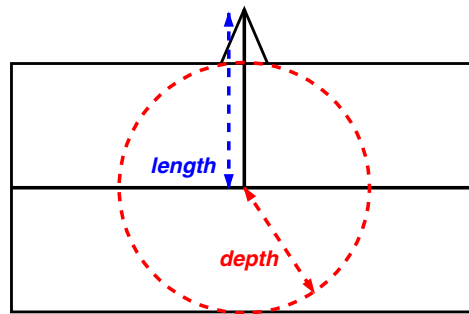


Fig. 4. Centerline pruning based on the branch length and the distance-from-surface (depth) information. A side branch is to be deleted if it is shorter than t_l and its branch-point is not closer to the border/surface than t_d .

starts from a branch whose diameter is larger than a given threshold t_d . Our method with repeated pruning steps (applying different pairs of thresholds (t_l, t_d)) is more general and can also delete entire unwanted subtrees.

The following algorithm is applied:

- (1) Calculate the linear time (3,4,5)—Chamfer distance map [3] for the elongated tree (after topological correction) in which the feature points are formed by the “0” voxels 6-adjacent to a “1” voxel. The resulting “distance-from-surface” map D is a non-binary array containing the distance to the closest feature voxel (Fig. 5a).

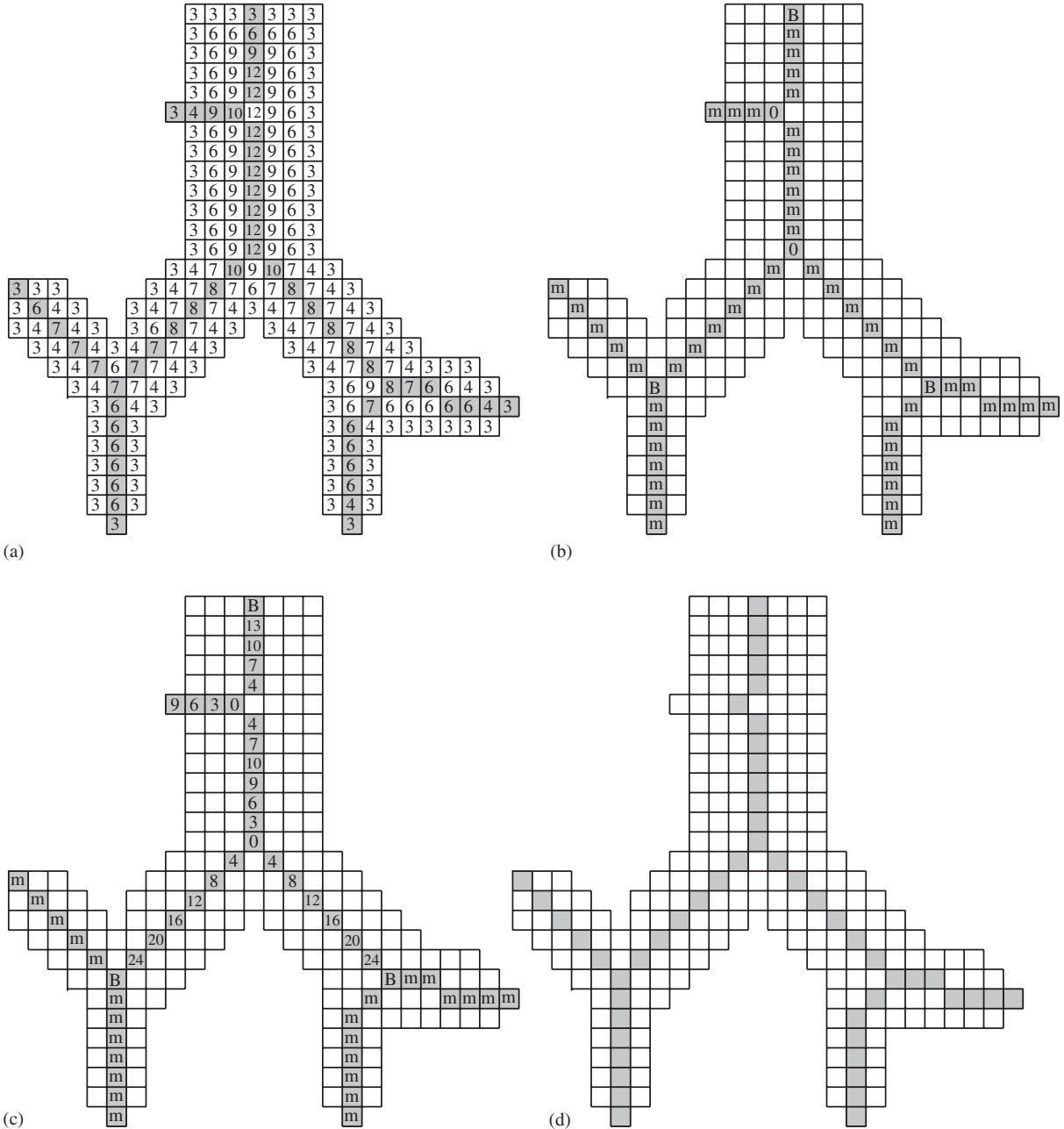


Fig. 5. Deletion of unwanted side branches. The pruning step is driven by the pairs of thresholds $(l_i, d_i) = (10, 10)$. For simplicity, a 2D tree and (3,4)-Chamfer distance are considered. (a) “Distance-from-surface” map in which darker pixels belong to the centerlines. (b) Initialization of the skeletal distance map. (c) Chamfer distance map. (d) Resulting centerlines after branch deletion.

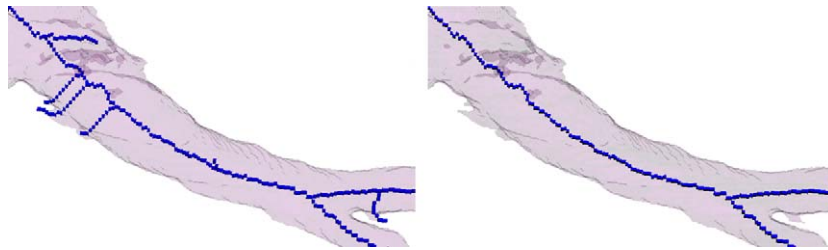


Fig. 6. A part of a segmented tree and its centerline before pruning (left) and after pruning (right). The applied pruning technique deletes unwanted long branches from thick parts and unwanted shorter ones from thinner parts, while correct branches are typically preserved throughout the tree.

(2) Initialization of the “skeletal distance map” SD :

$$SD(v) = \begin{cases} 0 & \text{if } v \text{ is a branch-point and } D(v) \geq t_d, \\ B & \text{if } v \text{ is a branch-point and } D(v) < t_d, \\ B & \text{if } v \text{ is the root of the tree,} \\ m & \text{otherwise,} \end{cases}$$

where values B and m should be larger than the maximal length in the tree. The B voxels are “bumpers” during the forthcoming distance-propagation step while the m voxels (assigned to line-points and endpoints in the centerlines) are to be changed (Fig. 5b).

(3) Distance propagation in SD according to the (3,4,5)—Chamfer distance can be performed similarly to the linear time Chamfer distance mapping (Fig. 5c). Note that the B voxels remain unchanged during this step.

(4) Branch deletion: a side branch with an associated endpoint v is deleted if $SD(v) \leq t_1$ (Fig. 5d). It can be done easily by using the Dijkstra’s tree or in the following way:

```

for  $i = t_1$  downto 1 do
  for each endpoint  $v$  in the centerlines do
    if  $SD(v) = i$  then
      delete  $v$  from the centerlines
  
```

Steps 2–4 of the above process can be repeated k times for different pairs of thresholds: (t_{1_1}, t_{d_1}) , (t_{1_2}, t_{d_2}) , \dots , (t_{1_k}, t_{d_k}) . In our experience, two to four iterations typically provide satisfactory results for in vivo airway trees. Note, that the thresholds are to be experimentally determined. The result of our pruning is demonstrated in Fig. 6.

2.6. Smoothing

The pruned centerline may be rough, therefore, a smoothing step is applied. It alters some one-, two-, and three-voxels long sequences of line-points. The smoothing rules are given by a set of matching templates. For brevity, only the templates capable of altering one voxel long skeletal segments are presented (Fig. 7).

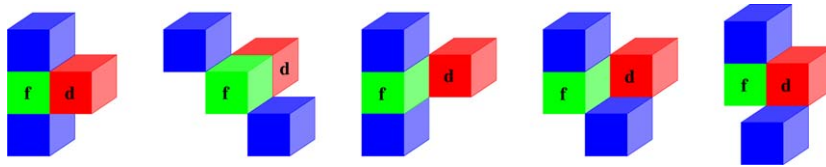


Fig. 7. The five base matching templates of our smoothing process capable of altering one voxel long parts. If the two blue voxels and the red one are line-points in the centerlines, then the red voxel is deleted (i.e., changed to “0”) and the non-skeletal green voxel is filled (i.e., changed to “1”). Note, that all rotations about the coordinate axes (where the rotation angles are 90° , 180° , and 270°) are also matching templates.

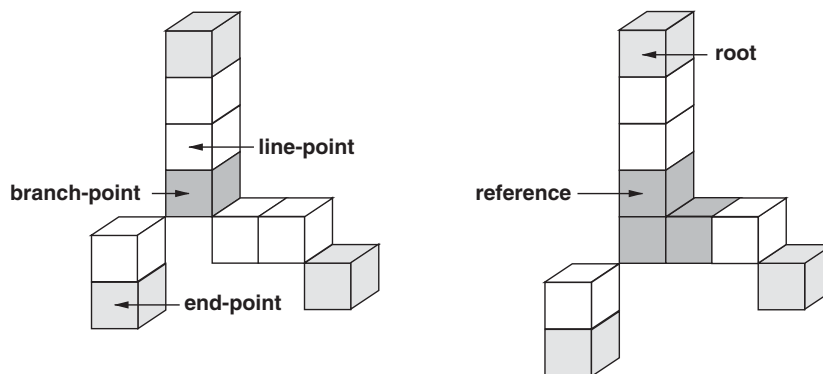


Fig. 8. The three types of voxels in a maximally thinned structure (left). Sometimes, a junction is formed by more than one branch-point (right). In that case, the branch-point closest to the root is the reference point of the junction.

2.7. Branch-point identification

In a skeleton, three types of points can be identified: endpoints (which have only one 26-neighbor), line-points (which have exactly two 26-neighbors), and branch-points (which have more than two 26-neighbors) that form junctions (bifurcations, trifurcations, etc.), see Fig. 8. Clearly, branch-identification from maximally thinned centerlines of an elongated tree is trivial. One problem is that more than one branch-point may form a junction. In that case, the branch-point closest to the root of the tree is assigned the junction label.

2.8. Generating formal tree structure

The formal tree structure assigned to the pruned centerlines is based on the updated Dijkstra’s tree (after pruning). It is stored in an array of n elements for a centerline containing n voxels. Each element of that array stores the coordinates of a voxel, its depth in the volume, and the index of the element that corresponds to the parent/predecessor voxel in the tree. This internal data structure is suitable for the forthcoming analysis and measurements, and provides an efficient coding of the resulting binary image. A similar structure is assigned to the branch-points. In the formal tree, a path between two branch-points is replaced by a single edge (Fig. 9).

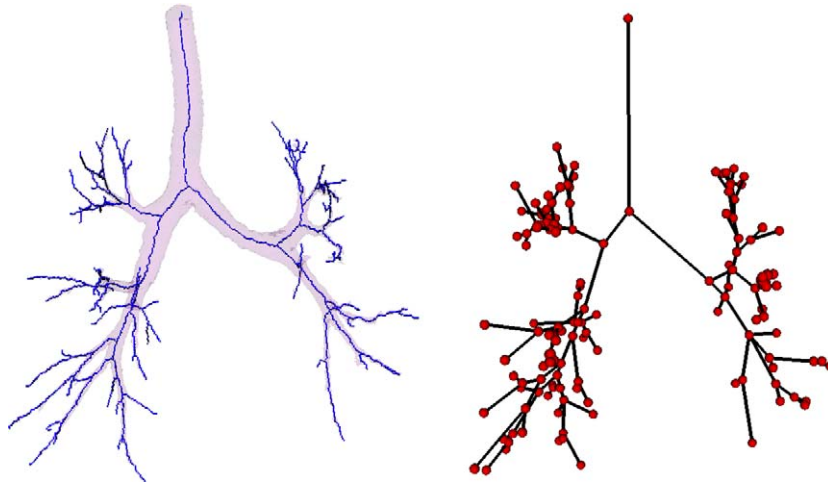


Fig. 9. A human airway tree and its centerlines (left). The corresponding formal tree structure, in which a path between two branch-points is replaced by a single edge (right).

2.9. Tree partitioning

The aim of the partitioning procedure is to partition all voxels of the segmented tree into branches—each voxel is assigned a branch-specific label. There are two inputs into the process—the segmented tree after topological corrections and the formal tree structure corresponding to the centerlines. The output is a gray-level image, in which value “0” corresponds to the background and different non-zero values are assigned to the voxels belonging to different tree branches/partitions.

The automated partitioning consists of two steps (Fig. 10). First, only the voxels in the centerlines are partitioned so that each branch/partition of the centerlines has a unique label. Non-skeletal tree voxels are then partitioned by label propagation—each voxel in the tree gets the label of the closest skeletal point.

The first step (i.e., centerlines partitioning) uses a queue Q (first-in-first-out data structure) and assigns label $l(v)$ and generation number $g(v)$ to the skeletal voxel v . That process is outlined as follows:

```

label = starting_label;
g(tree_root) = 1;
ENQUEUE( $Q$ , tree_root);
while NONEMPTY( $Q$ ) do
   $v$  = DEQUEUE( $Q$ );
   $l(v)$  = label;
  generation =  $g(v)$ ;
  while  $v$  has only one child do
     $u$  = ONLYCHILD( $v$ );
     $l(u)$  = label;
     $g(u)$  = generation;
     $v$  =  $u$ ;
endwhile
  
```

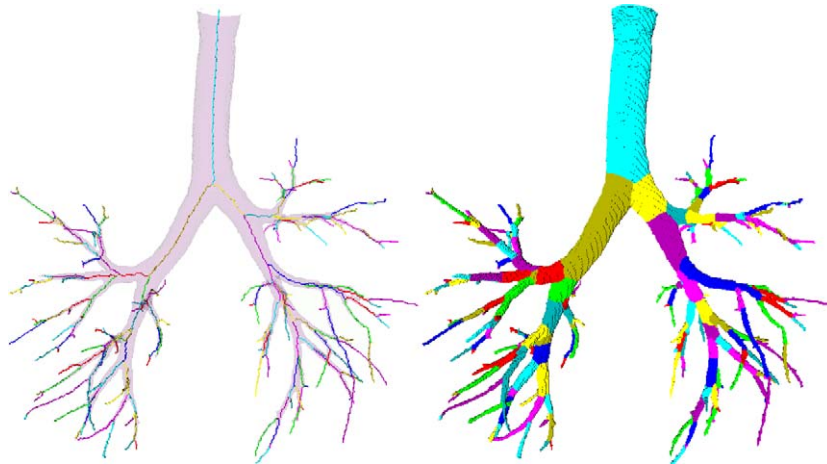


Fig. 10. Partitioning process: the segmented volume and the partitioned skeletal tree (left) and the partitioned volume after label-propagation (right). (Note that we used only nine colors in displaying these trees, therefore, the same color was assigned to multiple branches.)

```

label = NEXT(label);
generation = generation + 1;
for each child  $u$  of  $v$  do
     $g(u) = \text{generation}$ ;
    ENQUEUE( $Q, u$ );
endfor
endwhile

```

It works as follows. At first, the starting label is set and queue Q is initialized to contain the root of the skeletal tree. Each time through the outer **while** loop, a vertex v (i.e., the root or a branch-point) is extracted from the queue and labeled by the current label. Then all vertices (i.e., skeletal line-points) along the path from vertex v to the next branch-point are labeled by the same value (see the inner **while** loop). After labeling an entire branch, the current label is modified and the starting points of all adjacent branches are placed at the tail of the queue (see the **for** loop).

The algorithm—as implemented—is more difficult than as was presented. To deal with trifurcations, it is also capable of merging two branch-points if the distance between them is less than a given threshold. Trifurcations routinely appear in human airway trees and it is represented by a sequence of two bifurcations in a close proximity along the identified centerline.

2.10. Calculating associated measures

For each partition/branch of the tree, the following measures/indices are calculated:

- branch length—defined as a Euclidean distance between the parent and child branch-points (in mm),
- branch volume—defined as a volume of all voxels belonging to the branch (in mm^3),



Fig. 11. “Cylindrical” partitions in a human airway tree.

- branch surface area—defined as a surface area of all boundary voxels belonging to the branch (in mm^2),
- branch radius—derived from the branch length and the branch volume assuming “cylindrical” partition (in mm):

$$\text{radius} = \sqrt{\frac{\text{volume}}{\pi \text{ length}}}.$$

Determining the first three indices is fairly straightforward, but calculating a reliable approximation to the branch radius is rather complicated. The two ends of a partition/branch are “conic”, therefore, they must be suppressed to get measurements only from the “cylindrical” partitions.

First, skeletal voxels are re-labeled as follows:

$$l(v) \begin{cases} l(v) + \text{max_label} & \text{if voxel } v \text{ belongs to a “branch-point area”,} \\ l(v) & \text{otherwise,} \end{cases}$$

where $l(v)$ is the original label of the voxel v and max_label is greater than the largest label assigned to skeletal voxels during the centerlines partitioning process (therefore, $l(v) + \text{max_label}$ is a brand new label). “Branch-point areas” (i.e., the set of skeletal voxels to be re-labeled) are determined by the values in the distance map corresponding to the branch-points.

During the label propagation step (i.e., when each voxel in the tree gets the label of the closest skeletal point), the new labels are propagated. Determining branch volume and branch surface is based on labels $l(v) \bmod \text{max_label}$. Calculating the branch radius is based on “cylindrical” branch volume and “cylindrical” branch surface. Those indices are determined by considering labels being less than

max_label. (Note that branch radius is derived from the “cylindrical” branch length that is shorter than the calculated branch length.) The method is illustrated in Fig. 11.

3. Experimental methods

Performance of the reported method was assessed in 343 computer phantom instances subjected to changes of its orientation, in a rubber plastic phantom CT-scanned under 9 orientations, and in 6 in vivo scans of human lungs.

3.1. Phantoms

The computer phantom [35] is a 3D structural model of the human airway tree. The model consists of 125 elongated branches and its centerlines have 62 branch-points and 64 endpoints (including the root of the tree)—all positions of the branch-points are known. The generated object is embedded in a $300 \times 300 \times 300$ binary array containing unit-cube voxels. Independently, the phantom was rotated in 5° steps between -15° and $+15^\circ$ along all three axes.

The second phantom is a hollow rubber plastic one, casted from a normal human bronchial tree and consists of about 400 branches and 200 branch-points. The physical phantom was embedded in potato flakes (simulating lung tissue) and imaged in nine orientations using multi-row detector computed tomography with voxel size of $0.488 \times 0.488 \times 0.5$ mm (4-slice spiral CT, Mx8000, Philips Medical Systems). The volume sizes were $512 \times 512 \times 500 - 600$ voxels. The rotation angles defined nine phantom orientations in the scanner, the orientations were separated by 15° intervals in the $x - z$ and $y - z$ planes.

From 3D CT phantom images, segmentation was performed to separate bronchial airways from the lung parenchyma yielding a binary image used as an input to the reported skeletonization algorithm. For each of the $342 + 9 = 351$ phantom trees, skeletonization was performed fully automatically and the resulting skeletons were not interactively edited. For each instance of the computer phantoms, the branch-point position error was determined. It was defined as a Euclidean distance between the skeletonization-determined and true coordinates of the corresponding branch-points.

For a subset of nine computer phantoms and the nine rubber phantoms, the following quantitative indices were determined for the first five generations of the matched trees. Here, the reproducibility was determined by assessing differences between the reference tree and the tree analyzed in different orientations, after registering the analyzed tree with the reference tree. The quantitative measurements described above were compared in different orientations.

3.2. In vivo CT scans

The method was tested in six airway trees originating from in vivo scans of the human chest. For each subject, a scan close to total lung capacity (TLC) was acquired by multi-detector row spiral computed tomography with voxel size $0.683 \times 0.683 \times 0.6$ mm³ (4-slice spiral CT, Mx8000, Philips Medical Systems). The volume sizes were $512 \times 512 \times 450 - 550$ voxels. The segmented in vivo trees are not isometric (i.e., the voxels are cuboids instead of cubes) and contain numerous “thin” branches. To facilitate reproducibility assessment, a reference tree was constructed from each in vivo scan by removing such

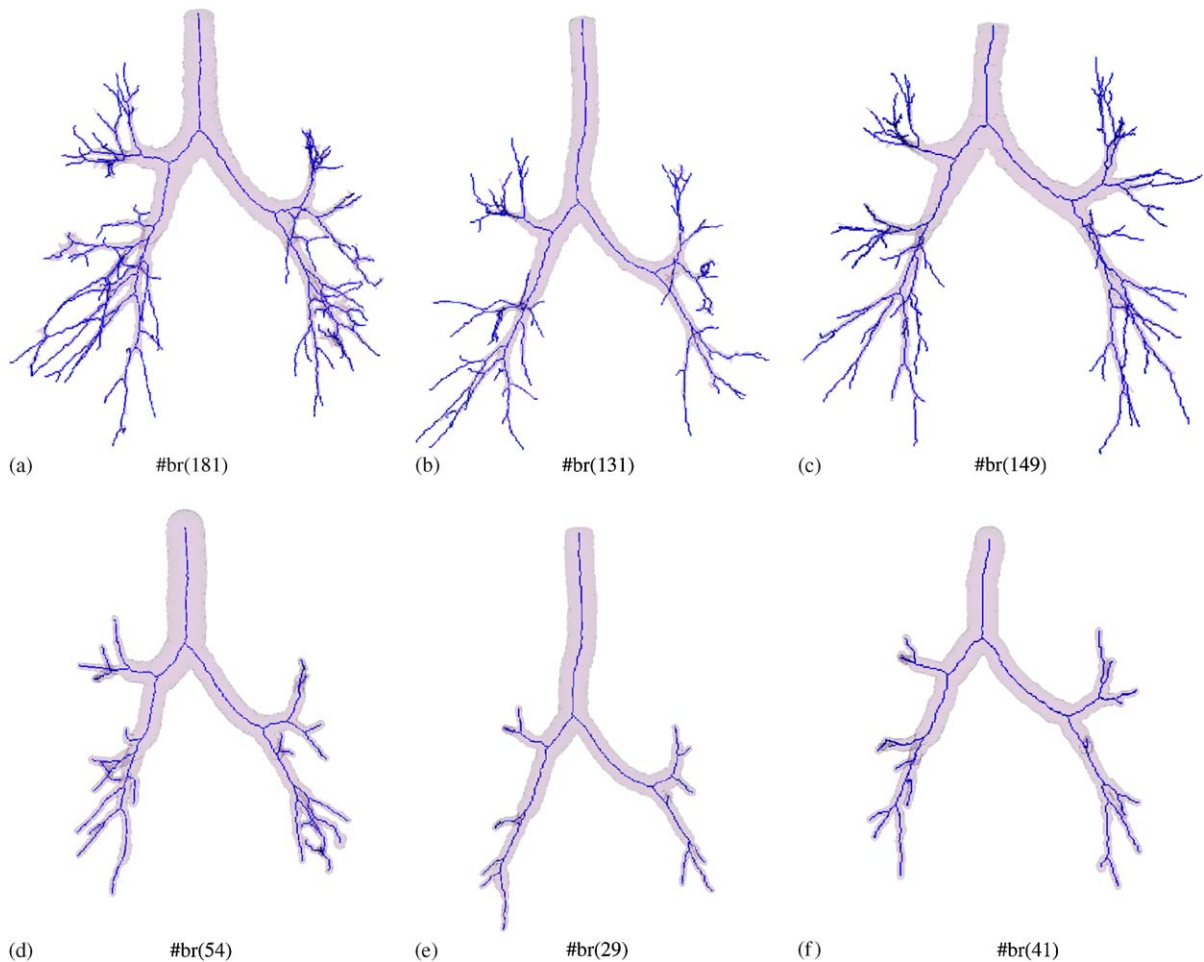


Fig. 12. Example of three in vivo TLC trees (a–c) and the corresponding reference trees (d–f) (#br means the number of branches).

thin branches and interpolating to achieve isometric cubic voxels (1 mm^3). The isometric reference trees were artificially rotated and used for reproducibility analysis. Fig. 12 gives an example of three of the original and reference trees and their centerlines.

For each reference tree, eight rotated instances were created. The orientations were separated by 15° intervals in the $x - z$ and $y - z$ planes. Skeletonization and quantitative analysis were performed fully automatically in 9×6 reference tree instances (six in vivo scans, nine reference trees per scan in neutral position plus eight rotations). The same parameter setting was used for all 54 trees. The quantitative measurements described above were compared for different orientations.

3.3. Statistical assessment

The reproducibility results are reported separately for the two phantom studies and for in vivo data. The average branch-point positioning errors are only calculated for the computer phantom for which

Table 1

Reproducibility of branch radius, branch length, branch volume, and branch surface area in computer phantoms, physical phantoms, and in vivo CT scans

Validation index	Computer phantoms	Physical phantoms	In vivo CT scans
Branch radius—bias	1.57E-16 voxel	−3.84E-18 mm	−3.39E-17 mm
95% interval of differences	(−0.0087, +0.0087)	(−0.0080, +0.0080)	(−0.0120, +0.0120)
Branch length—bias	−4.84E-16 voxel	1.39E-16 mm	−1.23E-16 mm
95% interval of differences	(−0.0797, +0.0797)	(−0.0987, +0.0987)	(−0.0497, +0.0497)
Branch volume—bias	−0.0052%	−0.0737%	−0.0464%
95% interval of differences	(−0.3747, +0.3643)	(−1.0424, +0.8951)	(−0.4510, +0.3582)
Branch surface—bias	−0.0198%	−0.0416%	0.0129%
95% interval of differences	(−0.3819, +0.3423)	(−0.9395, +0.8563)	(−0.2727, +0.2984)

the true branch-point positions were known. These errors are presented as mean \pm standard deviation and reported in voxels. All other reproducibility indices were compared using Bland–Altman statistic for which the average value of all corresponding measurements was used as an independent variable. The reproducibility showing 95% confidence intervals are presented in the form of Bland–Altman agreement plots [36].

To demonstrate the role of endpoint re-checking, the number of false branches before applying the pruning step was compared in 34 human airway trees. For each of the 34 trees, the number of branch-points was determined and compared for $t = 1$ (i.e., a new endpoint is to be deleted if it has at least one deleted 6-neighbor) and $t = 6$ (i.e., the endpoint re-checking is not activated).

4. Results

In the computer phantoms, the average branch-point positioning error showed subvoxel accuracy of 0.93 ± 0.41 voxel size. The reproducibility of the quantitative tree morphology indices demonstrated a minimal bias and a tight distribution of the branch diameter and length measurements. Reproducibility results are summarized in Table 1. The reproducibility of the quantitative tree morphology indices in in vivo CT scans are given in Fig. 13. In all cases, the relatively large differences between the surface and volume indices are to be expected due to a high sensitivity of these measures to minor partitioning errors, especially in short branches. Compare with the high reproducibility of the branch diameter and length measures. When assessing the re-checking performance in vivo, a statistically significant decrease of false branches was seen when the re-checking method was applied ($p \ll 0.001$).

5. Discussion

The reported approach to extracting centerlines from tree structures has several advantageous properties. It produces geometrically correct centerlines due to the employed directional approach (i.e., the outermost layer of an object is peeled by six successive sub-iterations according to the six main directions). As a result, the centerline is in its correct position (i.e., in the middle of the object) and its location is fairly invariant under object orientation as demonstrated above.

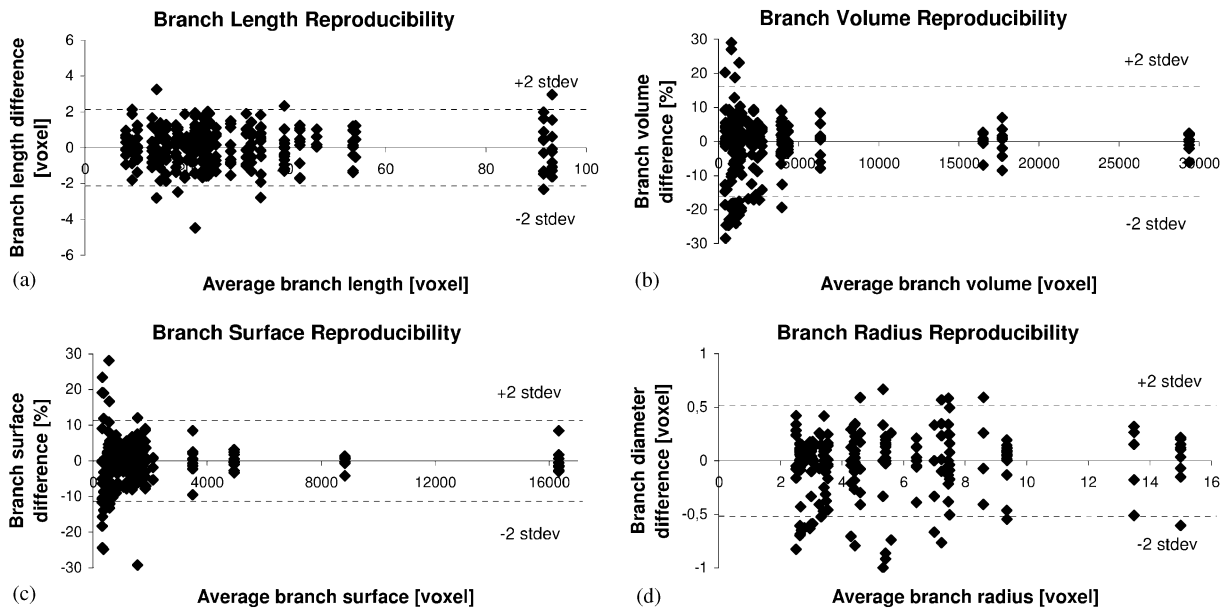


Fig. 13. Reproducibility in the 54 “reference” trees derived from in vivo CT data: (a) branch length, (b) branch volume, (c) branch surface area, and (d) average branch radius.

The produced centerline is topologically equivalent to the original elongated object, since simple points are deleted sequentially.

Our algorithm produces maximally thinned (i.e., 1-voxel wide) centerlines, since all simple points are deleted. In consequence, each of our skeletal trees is minimal; if we delete any non-endpoint, then the structure would be disconnected. Note, that some thinning algorithms may delete only a subset of simple points [29], therefore, some “kinks” (i.e., segments with more than 1-voxel wide) or some 2-voxels-wide surface patches may be present in their centerlines.

Our skeletonization retains the shape of the original (elongated) object by preserving line-end points. The endpoint preserving thinning differs from shrinking (it is capable of extracting the topological kernel of an object). Our approach creates a substantially smaller number of unwanted centerline segments compared to competing methods [31] due to the novel endpoint re-checking step. While branches can typically be removed by pruning, the proposed re-checking represents a problem prevention strategy since it has the following advantageous properties: (1) When re-checking is incorporated, a much smaller number of branches need to be removed during the pruning step. As a result, the pruning process is faster and pruning parameters are easier to set. (2) It is less unlikely that unwanted side branches form sub-trees. (3) A side branch may disturb the centerline around its corresponding branch-point. That kind of disturbance cannot be corrected by the pruning process. The proposed re-checking process is based on counting the deleted 6-neighbors of the marked points and it statistically decreases the number of identified branch- and endpoints without removing valid branches ($p \ll 0.001$). This count depends on the object orientation. Consequently, counting all the neighbors may possibly be more efficient. Similarly, the result of our thinning method depends on the order of the deletion directions and the scanning order

on the points. Investigating alternative re-checking techniques and different orders of deletion directions and scanning is considered as a further research work.

Some authors (e.g., Tsao and Fu [2] or Nyström [7]) proposed a 2-step method to extract centerlines (medial lines) from 3D objects: first, surface-skeleton is produced as an intermediate result, then the surface-skeleton is reduced to a curve-skeleton (centerlines). It is shown in [29], that the 2-step method for centerlines extraction is more sensitive to coarse object boundaries than the direct methods. Consequently, more unwanted parasitic line segments are created. Therefore, our thinning algorithm directly extracts centerlines by preserving only the line-end points.

Additionally, our method allows an easy and efficient implementation. Two linked lists are used; the first one stores all border points and the second one stores all border points of the actual type that are simple and not line-end points. The simplicity of a point is decided by determining a 26-bit integer code corresponding to their $3 \times 3 \times 3$ neighborhood and addressing a pre-calculated (unit time access) look-up-table containing all possible $3 \times 3 \times 3$ configurations. The codes of the marked points are stored on the second list. Therefore, the endpoint re-checking can be performed easily. Whenever a point is deleted, it is removed from the first list and its 6-neighbors (which are not on that list) are added. By using linked lists, the expensive scanning of the volume is avoided. The proposed thinning algorithm is fairly fast; our implementation takes only 7 s for a $512 \times 512 \times 512$ image containing 250,000 object voxels (running Linux on a 1.7 MHz AMD Athlon PC) including reading the input volume and the 8 MB look-up-table, and writing the output image. It is much faster than the parallel thinning algorithms employed, e.g., in [8].

Our labeling method uses a generation-by-generation tree-walk strategy. Therefore, it can handle any kinds of trees including trees containing trifurcations, four-furcations, etc. In comparison, the labeling method proposed by Mori et al. [12] may fail if a trifurcation is present. Note that during the labeling process, an expanded branch-tree structure containing generation numbers can be created. Additional quantitative indices including local radius of a tree branch, bifurcation angle, etc. can be calculated. Note, that computing a reliable branch surface area is not trivial. The calculated surface area (i.e., sum of all facet surfaces) does not converge toward the true surface area for an increasing resolution. Mori et al. [5] represented the surface data of the bronchus wall by the Marching Cubes method. They calculated the minimum distance between the center of each triangle patch and the branch path. If the distance was lower than a given threshold value, that patch was assigned to the branch. A set of triangles belonging to the same branch formed the surface of that branch. Note, that their approach allows duplicate classification.

While very powerful, the presented approach is not free of several limitations. (1) The current method works only for scans containing a single tree, although it can be modified to work with set of trees. (2) As we discussed earlier, the tree root is currently identified automatically using a priori airway-tree-specific knowledge—the root identification need to be modified for other applications. (3) Some parameters (including the pruning thresholds and the maximal distance for which two bifurcations shall be replaced with a trifurcation) are problem dependent and are currently set manually. Automated setting is deemed feasible.

6. Conclusion

The presented automated method for skeletonization, branch-point identification and quantitative analysis of tubular tree structures is robust, efficient, and highly reproducible. It facilitates calculation of a

number of morphologic indices described above as well as indices not considered in this work—branch angle, curvature, and many others.

The developed approach builds on the following concepts and properties: fast curve-thinning algorithm to increase computational speed, endpoint re-checking to avoid generation of spurious side branches, depth-and-length sensitive pruning, exact tree-branch partitioning allowing branch volume and surface measurements, identification of non-branching tree segments, achieving sub-voxel accuracy of branch point positioning, and performing extensive validations on complex phantoms and in vivo scans.

Acknowledgements

This work was supported by the NIH Grant HL-064368. The authors would like to thank Jered Sieren for performing a large number of multiple-angle scans of phantoms used in the validation studies.

References

- [1] G. Gerig, T. Kollera, G. Székely, C. Brechbuhler, O. Kubler, Symbolic descriptions of 3-D structures applied to cerebral vessel tree obtained from MR angiography volume data, in: *Proceedings of 13th International Conference on Information Processing in Medical Imaging, IPMI'93*, ser. *Lecture Notes in Computer Science*, vol. 687, Springer, Berlin, 1993, pp. 94–111.
- [2] Y. Tsao, K. Fu, A parallel thinning algorithm for 3-D pictures, *Comput. Graphics Image Process.* 17 (1981) 315–331.
- [3] G. Borgefors, Distance transformations in arbitrary dimensions, *Comput. Vision Graphics Image Process.* 27 (1984) 321–345.
- [4] S. Wood, A. Zerhouni, J. Hoford, E. Hoffman, W. Mitzner, Measurement of three-dimensional lung tree structures using computed tomography, *J. Appl. Physiol.* 79 (1995) 1687–1697.
- [5] K. Mori, J. Hasegawa, J. Toriwaki, H. Anno, K. Katada, A fast rendering method using the tree structure of objects in virtualized bronchus endoscope system, in: *Proceedings of 4th International Conference on Visualization in Biomedical Computing, VBC'96*, ser. *Lecture Notes in Computer Science*, vol. 1131, Springer, Berlin, 1996, pp. 33–42.
- [6] T. Saito, J. Toriwaki, A sequential thinning algorithm for three dimensional digital pictures using the Euclidean distance transformation, in: *Proceedings of 9th Scandinavian Conference on Image Analysis*, ser. *SCIA'95*, 1995, pp. 507–516.
- [7] I. Nyström, Skeletonization applied to magnetic resonance angiography images, in: *Proceeding of Medical Imaging 1998: Image Processing*, ser. *SPIE*, vol. 3338, 2003, pp. 693–701.
- [8] S. Wan, A. Kiraly, E. Ritman, W. Higgins, Extraction of the hepatic vasculature in rats using 3-D micro-CT images, *IEEE Trans. Medical Imaging* 19 (2000) 964–971.
- [9] P. Saha, B. Chaudhury, D. Majumder, A new shape-preserving parallel thinning algorithm for 3D digital images, *Pattern Recognition* 30 (1997) 1939–1955.
- [10] S. Wan, E. Ritman, W. Higgins, Multi-generational analysis and visualization of the vascular tree in 3D micro-CT images, *Comput. Biol. Med.* 32 (2002) 55–71.
- [11] K. Krissian, G. Malandain, N. Ayache, R. Vaillant, Y. Troussset, Model based detection of tubular structures in 3D images, *Comput. Vision Image Understanding* 80 (2000) 130–171.
- [12] K. Mori, J. Hasegawa, Y. Suenaga, J. Toriwaki, Automated anatomical labeling of the bronchial branch and its application to the virtual bronchoscopy system, *IEEE Trans. Medical Imaging* 19 (2000) 103–114.
- [13] J. Toriwaki, K. Mori, Distance transformation and skeletonization of 3D pictures and their applications to medical images, in: *Digital and Image Geometry*, ser. *Lecture Notes in Computer Science* 2243, Springer, Berlin, 2001, pp. 412–429.
- [14] T. Sera, H. Fujioka, H. Yokota, A. Makinouchi, R. Himeno, R. Schroter, K. Tanishita, Three-dimensional visualization and morphometry of small airways from microfocal X-ray computed tomography, *J. Biomech.* 36 (2003) 1587–1594.
- [15] S. Aylward, E. Bullitt, Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction, *IEEE Trans. Medical Imaging* 21 (2002) 61–75.

- [16] M. Maddah, A. Afzali-Kusha, H. Soltanian-Zadeh, Efficient center-line extraction for quantification of vessels in confocal microscopy images, *Med. Phys.* 30 (2003) 204–211.
- [17] Z. Chen, S. Molloi, Automatic 3D vascular tree construction in CT angiography, *Computerized Med. Imaging Graphics* 27 (2003) 469–479.
- [18] K. Palágyi, A. Kuba, A 3D 6-subiteration thinning algorithm for extracting medial lines, *Pattern Recognition Lett.* 19 (1998) 613–627.
- [19] T. Deschamps, L.D. Cohen, Fast extraction of minimal path in 3D images and applications to virtual endoscopy, *Med. Image Anal.* 5 (2001) 281–299.
- [20] S. Bouix, K. Siddiqi, A. Tannenbaum, Flux driven fly throughs, in: *International Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Madison, WI, 2003, pp. 449–454.
- [21] L. Antiga, B. Ene-Iordache, A. Remuzzi, Computational geometry for patient-specific reconstruction and meshing of blood vessels from angiography, *IEEE Trans. Medical Imaging* 22 (2003) 674–684.
- [22] J.K. Udupa, S. Samarasekera, Fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation, *Graphics Models Image Process.* 58 (1996) 246–261.
- [23] G.T. Herman, B.M. Carvalho, Multiseeded segmentation using fuzzy connectedness, *IEEE Trans. Pattern Anal. Machine Intell.* 23 (5) (2001) 460–474.
- [24] T. Kitasaka, K. Mori, H.-i. Hasegawa, Y. Suenaga, J.-i. Toriwaki, Extraction of bronchus regions from 3D chest X-ray CT images by using structural features of bronchus, in: *Computer Assisted Radiology and Surgery (CARS) 2003*, International Congress Series 1256, Elsevier, Amsterdam, 2003, pp. 240–245.
- [25] J. Tschirren, segmentation, anatomical labeling, branch-point matching, and quantitative analysis of human airway trees in volumetric CT images, Ph.D. dissertation, University of Iowa, IA, USA, 2003.
- [26] R. Gonzales, R. Woods, *Digital Image Processing*, Addison-Wesley, Reading, MA, 1992.
- [27] M. Naf, G. Székely, R. Kikinis, M. Shenton, G. Kubler, 3D Voronoi skeletons and their usage for the characterization and recognition of 3D organ shape, *Comput. Vision Graphics Image Process.* 66 (1997) 147–161.
- [28] T. Kong, A. Rosenfeld, Digital topology: introduction and survey, *Comput. Vision Graphics Image Process.* 48 (1989) 357–393.
- [29] K. Palágyi, A. Kuba, A parallel 3D 12-subiteration thinning algorithm, *Graphical Models Image Process.* 61 (1999) 199–221.
- [30] G. Malandain, G. Bertrand, Fast characterization of 3D simple point, in: *Proceedings of 11th IEEE International Conference on Pattern Recognition*, 1992, pp. 232–235.
- [31] T. Lee, R. Kashyap, C. Chu, Building skeleton models via 3-D medial surface/axis thinning algorithms, *CVGIP: Graphical Models Image Process.* 56 (1994) 462–478.
- [32] D. Shaked, A. Bruckstein, Pruning medial axes, *Comput. Vision Image Understanding* 69 (1998) 156–169.
- [33] S. Svensson, G.S. di Baja, Simplifying curve skeletons in volume images, *Comput. Vision Image Understanding* 90 (2003) 242–257.
- [34] T. Cormen, C. Leiserson, R. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1990.
- [35] H. Kitaoka, R. Takaki, B. Suki, A three-dimensional model of the human airway tree, *J. Appl. Physiol.* 87 (1999) 2207–2217.
- [36] J.M. Bland, D.G. Altman, Statistical methods for assessing agreement between two methods of clinical measurement, *Lancet* 1 (8476) (1986) 307–310.

Kálmán Palágyi (Ph.D. 2000, University of Szeged) is Associate Professor at the Department of Image Processing and Computer Graphics, Faculty of Science, University of Szeged. His research interests include thinning and skeletonization algorithms in 3D and their medical applications.

Juerg Tschirren (Ph.D. 2003, University of Iowa) is Director of Engineering at VIDA Diagnostics. His main research interest is medical image processing and during the last few years he focused on pulmonary CT image analysis.

Eric A. Hoffman is a professor of radiology and biomedical engineering at the University of Iowa, director of the Physiological Imaging Laboratory in the Department of Radiology, and director of the Iowa Comprehensive Lung Imaging Center (I-Clic) at the University of Iowa. He received his Ph.D. in Physiology from the University of Minnesota/Mayo Graduate School of

Medicine in 1981 and remained on staff at the Mayo Clinic where he was a member of the team which introduced volumetric CT imaging. Dr. Hoffman moved from the Mayo Clinic in 1987 to head the Cardiothoracic Imaging Research Center in the Department of Radiology at the University of Pennsylvania and then moved in 1992 to his current position at the University of Iowa.

Throughout his career, he has utilized non-invasive imaging methods to study the basic physiology of the heart and lungs using both CT and MRI imaging methods. Most recently he has begun to apply multidetector row spiral CT imaging methodology to quantitate human heart and lung pathology with a particular emphasis on utilizing X-ray CT to provide a comprehensive assessment of lung structure and function. Dr. Hoffman is the author of more than 200 journal articles and 20 book chapters in the field of dynamic volumetric imaging. Dr. Hoffman is a Fellow of the American Institute for Medical and Biological Engineering.

Milan Sonka received his Ph.D. degree in 1983 from the Czech Technical University in Prague. He is Professor of Electrical and Computer Engineering at the University of Iowa and IEEE Fellow.

Sonka's research interests include medical imaging and knowledge-based image analysis. A major focus of his research in the last several years has been development of clinically applicable automated techniques for cardiovascular analysis; pulmonary CT image analysis; cell tracking and cellular shape analysis, and augmented reality image-based surgical planning.

He is the first author of a book *Image Processing, Analysis and Machine Vision* published in 1993 by Chapman and Hall in London, 2nd edition 1998 by PWS, Pacific Grove, CA. He has co-authored or co-edited 10 other books including *Handbook of Medical Imaging, Volume II—Medical Image Processing and Analysis* published in 2000. He has authored 7 book chapters, over 60 journal papers, 160 conference papers, and 60 abstracts. He is Associate Editor of the *IEEE Transactions on Medical Imaging*, and member of the Editorial Board of the *International Journal of Cardiovascular Imaging*.