Simultaneous Segmentation and Tree Reconstruction of the Airways for Virtual Bronchoscopy

Thorsten Schlathölter[*a], Cristian Lorenz[**a], Ingwer C. Carlsen[a], Steffen Renisch[a], Thomas Deschamps [b]
[a]Philips Research Hamburg; [b]Philips Research France

ABSTRACT

During the last couple of years virtual endoscopic systems (VES) have emerged as standard tools that are nowadays close to be utilized in daily clinical practice. Such tools render hollow human structures, allowing a clinician to visualize their inside in an endoscopic-like paradigm. It is common practice that the camera of a virtual endoscope is attached to the centerline of the structure of interest, to facilitate navigation. This centerline has to be determined manually or automatically, prior to an investigation. While there exist techniques that can straightforwardly handle "simple" tube-like structures (e.g. colon, aorta), structures like the tracheobronchial tree still represent a challenge due to their complex branching. In these cases it is necessary to determine all branching points within the tree which is - because of the complexity - impractical to be accomplished in a manual manner. This paper presents a simultaneous segmentation/skeletonization algorithm that extracts all major airway branches and large parts of the minor distal branches (up to 7th order) using a front propagation approach. During the segmentation the algorithm keeps track of the centerline of the segmented structure and detects all branching points. This in turn allows the full reconstruction of the tracheobronchial tree.

Keywords: Virtual Bronchoscopy, endoscopy, segmentation, fast marching, front propagation, skeletonization, computed tomography, tree reconstruction, multi slice CT

1. INTRODUCTION

One of the recent trends in volumetric analysis of high-resolution CT data is virtual reality imaging, which is a form of interactive three-dimensional visualization that can provide computer-rendered, intraluminal views of any hollow structure [1]. This kind of visualization technique is also known as virtual endoscopy (VE). Virtual endoscopy offers the potential to navigate through computer simulations of the human body in a manner comparable to fiber-optic viewing. This makes it possible to study cases prior to a real endoscopic intervention, and it can therefore be of great help in the planning phase or in follow up studies. VES can be used to supply additional information to the physician because they can easily combine different kind of interactive image assessment, like surface renderings, volume renderings, endoscopic views, MIP/mIP, multi-planar reformattings, measurement data, etc.[2], and they can for example be directly connected to information databases or anatomical atlases. Together with computer aided diagnostic tools VES have the potential to become standard methods that will be frequently utilized in daily clinical practice. Therefore, the system must be easy to use and the time needed to analyze an individual case must be reasonably small. VES must be designed to support the physician as much as possible. One aspect in this sense is navigation. It is impractical to steer a virtual camera through the data set manually using input devices like the keyboard or the mouse. So the operator of a VES must be given the possibility to easily specify paths through the hollow structure of interest, which can then be used for fly-throughs of a virtual camera. These paths can also be used for fast access to arbitrary points within the structure. Current systems often offer the possibility of manual path definition. It is clear that this option is only feasible for very simple structures without too much branching (e.g. parts of vessels). The colon for example is tube-like without any branching, but path definition is nevertheless awkward because of its labyrinthine structure. Although it is still possible to define a path through the colon, it is too time consuming in clinical practice. To overcome this problem

* thorsten.schlathölter@philips.com; phone (+49) 40-5078-2051; fax (+49) 40-5078-2510; Philips Research Hamburg , Roentgenstr. 24-26, D-22335 Hamburg; **cristian.lorenz@philips.com; phone (+49) 40-5078-2063; fax (+49) 40-5078-2510; Philips Research Hamburg , Roentgenstr. 24-26, D-22335 Hamburg

Medical Imaging 2002: Image Processing, Milan Sonka, J. Michael Fitzpatrick,
Editors, Proceedings of SPIE Vol. 4684 (2002) © 2002 SPIE · 1605-7422/02/$15.00

several methods have been proposed to automatically determine the centerline of the colon[3]. When one is dealing with more complex structures comprising a lot of branching, automatic centerline detection becomes even more important. One example for such a structure is the human tracheobronchial tree, which has segments with a branching order of at least 6 that are still accessible by modern fiberoptic endoscopes. Manual centerline definition for the whole bronchial tree is nearly impossible and therefore it is essential to have a method at hand that automates this task. In the following we will call the set of centerlines describing a tree structure a skeleton. This skeleton could be a binary subset of the original data set of one pixel width. For practical purposes it is desirable to have access to all individual branches of the skeleton which means that all branching points within the skeleton have to be determined. We precisely specify a skeleton to be a 26-connected, one-voxel-thick representation of the tree that comprises all branching points and the respective connecting centerlines.

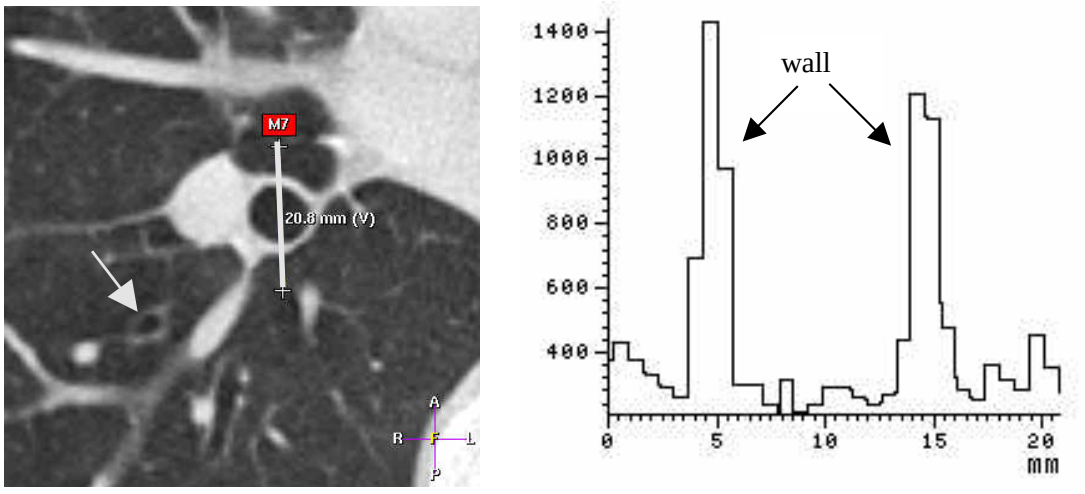
One strategy for the generation of skeletons is to use thinning algorithms on segmented structures with a subsequent refinement to determine branching points[4,5]. Another approach is to apply distance maps to the segmented structure and determine paths from the ends of the branches (the voxels with the largest distance) to the root branch[6]. Consecutive determination of all end-root-paths can finally lead to a skeleton according to the above specification. These approaches have in common that they start from an already existing segmentation of the object to be skeletonized. We have developed an algorithm which is, to our knowledge, unique in the sense that we determine the skeleton structure during the segmentation phase. So we have a one-step process. The major advantage of this one step process is that we can use the derived skeleton for the incorporation of model information into the segmentation process. This helps us to detect leakage of the grow process into the lung parenchyma - a well-known problem in bronchi segmentation - and it significantly improves our segmentation results.

2. BRONCHIAL ANATOMY

The human lung consists of two major parts, the left lung and the right lung. There are three lobes in the right lung, which are separated by the so-called major fissure and minor fissure. The left lung shows a slightly different structure. Because there is no defined minor fissure, it consists of only two lobes, whereby the part that anatomically corresponds to the right middle lobe is merged with the upper lobe. Each lobe is again divided into two or more lung segments of which ten exist for each side of the lung. These segments are supplied by a complex system of branching trees that conduct blood and air into the distal regions where the gas exchange takes place. The bronchial tree has a pipe structure that is filled with air. It starts at the trachea and extends into the distal regions repeatedly splitting into smaller and smaller branches. In the human lung, the splitting occurs usually in bifurcations, e.g. the parent branch splits up into two child branches, but trifurcations also exist. The general tendency for child branches is that they decrease in diameter and length although this might be different in individual cases. Siblings don't necessarily have the same diameter. The bronchi are classified into lobar bronchi (supplying the 5 lobes), segmental bronchi (supplying the 20 individual segments), and sub-segmental bronchi. The air-filled lumen of the bronchi is surrounded by the bronchial wall. The thickness of this wall is correlated to the diameter of the segment in the sense that it gets thinner for smaller diameters. High-resolution multi-slice CT reveals bronchi segments in the 6^{th} branching generation (and higher) which have diameters in the mm range. Bronchi walls of these sub-segmental bronchi become very thin so that the partial volume effect and image artifacts can break up the well-defined wall structure. This can cause severe problems in the segmentation of the bronchi as described in the next chapter.

3. SEGMENTATION AND SKELETONIZATION ALGORITHM

One of the strategies for extracting the airways is region growing. Since this technique uses a threshold to determine whether a voxel is included into the segmentation or not, there exists a severe problems with bronchi segmentation. The reason is depicted in Figure 1 showing a CT slice with a lobar bronchus cross-section in the center of the image. The lobar bronchus in this slice has a well-defined wall that separates the lumen from the surrounding parenchyma. Figure 1b shows the intensity profile across the line in Figure 1a. From this it is obvious that the average gray value of the bronchi lumen and the parenchyma differs only slightly. Thus a threshold that includes the bronchi lumen and that excludes the parenchyma cannot easily be chosen. At first view this does not represent a severe problem because the bronchial wall very well separates the two regions and segmentation should easily be accomplished by choosing a threshold value close to the bronchial wall. However, the wall becomes thinner for the sub-segmental bronchi and the partial volume effect results in a blurring of the wall, which finally leads to holes. This case is depicted in Figure 1a (arrow). If the segmentation process encounters such a hole, it will usually leak into the parenchyma. If it was such that leakage would occur at the end of the region growing when all other important structures have already been segmented,



a b

Figure 1: Segmentation of bronchial tree. (a) CT slice showing a segmental bronchus. The intensity profile across the bronchus is shown in (b) (arbitrary units). The intensity difference between bronchial lumen and parenchyma is very small, however the bronchial wall separates clearly lumen and parenchyma. Due to partial volume effect and image artifacts this barrier breaks down at smaller bronchial diameter (arrow).

the user could just interactively stop the grow process at that point, or this point could be derived automatically[7]. Unfortunately this is usually not the case and once leakage occurs the grow process is kept from growing still valid regions. There are different approaches to the problem of leakage. Summers et al for example proposed a method to overcome leakage by controlling the distance to a given seed point[8], Sonka et al. combine 3D region growing with rule-based algorithms that take a priori anatomical knowledge into account [9], and Law and Heng try to account for the problem by the determination of an optimal threshold value[7].

We propose a method that is able to detect leakage and that automatically stops growing leaked regions, thus allowing the complete segmentation of all valid regions. The technique is based on the front propagation method, which is used in conjunction with an anatomical model of the tracheobronchial tree. The front propagation method is a type of region growing technique that uses a concept motivated from physical wave-front propagation and that is based on the physical principle of least action. The front propagation equation is of the type:

$$|\nabla T| F = 1 \tag{1}$$

where $F(\mathbf{x})$ is the speed function of the front and $T(\mathbf{x})$ denotes the time value when the front reaches the point $\mathbf{x}$. To solve this equation we take advantage of the fast marching method that makes use of the fact that the front is always propagated from smaller values of T to larger ones[10]. Doing so the solution of equation (1) reduces to a quadratic equation to be solved for each voxel. If the speed function is made dependent on the image data, the fast marching method can straightforwardly be adapted to image segmentation[3]. In this paper we will concentrate on the modifications we made to the algorithm in order to cope with the problem of bronchi segmentation. We propose not to use an image-like grid of floating-point numbers to store the time-stamps, as it is usually done in current implementations. Since the bronchial tree comprises just a small portion of the overall data set, it is possible to replace the floating-point grid by a hash-table. The time-stamps are stored in the table using the voxel position within the grid as a key. In this way it becomes possible to deal with large data sets of for example 512^3 voxels. In addition to the higher memory efficiency, we also avoid the allocation and initialization phase of the floating-point grid, which can take in the order of seconds for large data sets. In the following the grow process is always seeded at the first slice of the data set showing the trachea. We applied the modified algorithm with an exponential speed-function of the form:

$$F(\mathbf{x}) = e^{-\beta|\nabla I(\mathbf{x})|}, \quad \beta > 0, \tag{2}$$

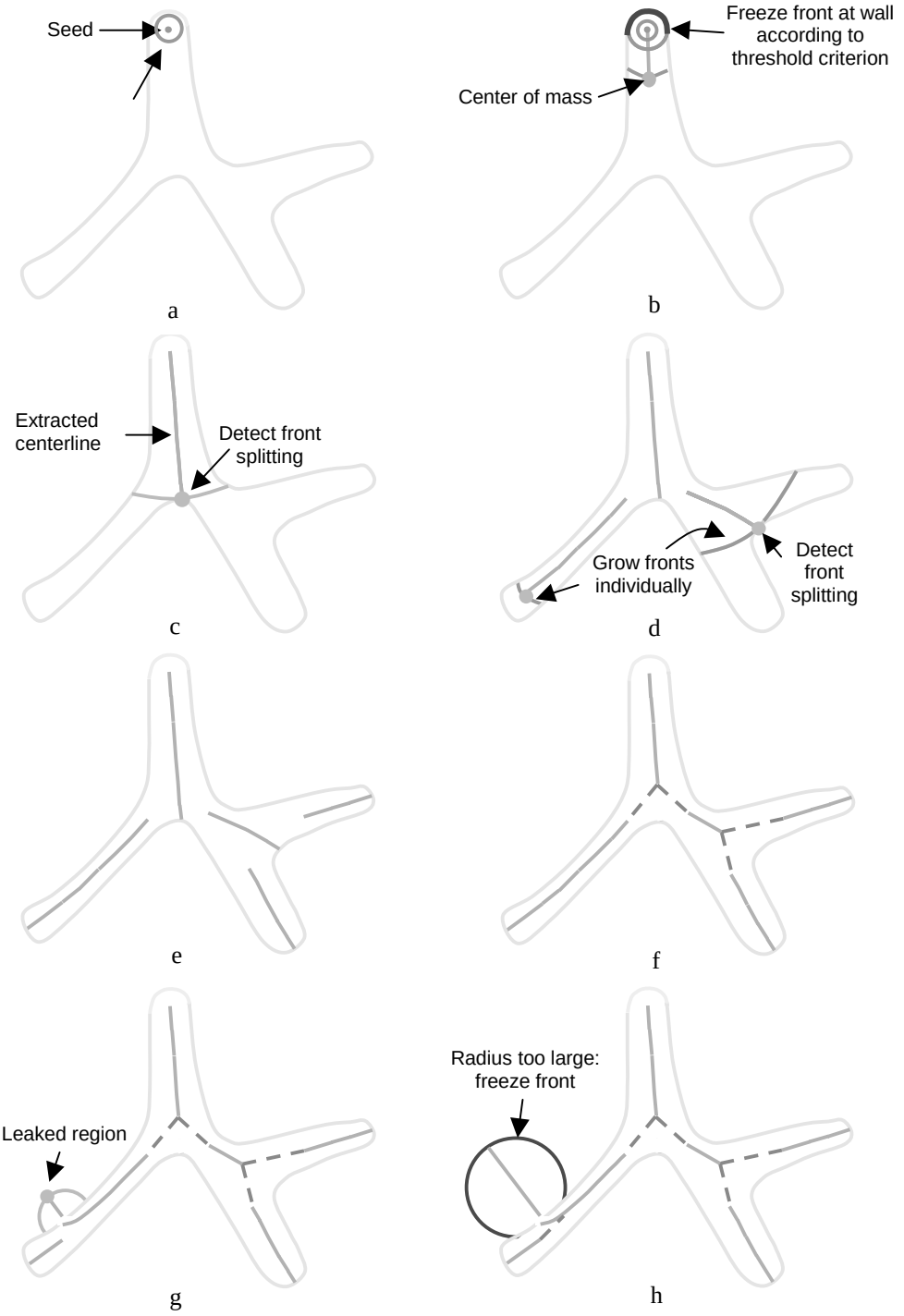


Figure 2**:** Illustration of the grow process. (a) place seed and grow front; (b) freeze front at wall and keep track of the center of mass; (c) detect splitting of front at bifurcation; (d) grow the individual parts of the front separately; (e) the raw skeleton as extracted by the algorithm; (f): refinement of the skeleton to reflect the correct branching structure; (g): leakage into the parenchyma; (h): freezing of front due to a radius criterion.

to a multi-slice CT data set with 512^3 voxels. In this equation $I(\mathbf{x})$ denotes the gray value at the point **x**. Function (2) obviously moves fast in homogeneous regions of the image and slowly in regions with a high value of image gradient, e.g. the bronchial wall. It turned out that the parameter β, could not be chosen such to prevent leaking into the parenchyma in all cases.

In a next step we chose a stepwise constant speed function of the following form:

$$F(x) = \begin{cases} 1 \text{ for } I(x) \le t \\ 0 \text{ for } I(x) > t \end{cases}, \tag{3}$$

with t being a threshold value just above the bronchial lumen. The result is similar to a simple region growing with the important difference that the front always moves at the points with the closest geodetical distance to the seed point. Running in a tube like structure the front behaves similar to the front of a viscous fluid in a pipe. At voxels that do not conform to the threshold value t, i.e. wall voxels, the front is practically frozen. Thus the active front does not represent a closed entity anymore that comprises the whole segmented region, as with the conventional fast marching algorithm. Instead, once the segmented region touches the wall, the front reduces to a centroid which has a surface normal in the direction of the branch axis (Figure 2b). At branching points the front splits into two or more centroids, each individually following a different child branch (Figure 2 c,d). This in turn allows the detection of branching by checking the connectivity of the grow-front. Practically this can be accomplished by a simple region growing algorithm that uses the top of the grow-front as a seed and that successively labels all connected voxels with an integer number. If all connected voxels have been labeled and there are still voxels in the front without a valid label, the integer number is increased and another grow process is started with one of the remaining unlabeled voxels as seed. The process is repeated until all front voxels have a valid label. In case two or more different labels result from the procedure, branching occurred. Each set of voxels can then be used to initialize and start an independent grow process. In addition to the extraction of all branch points, the described method is able to supply a good approximation of the centerline by keeping track of the centroids center of mass (Figure 2 c). After all children of a parent branch have been grown, the centerlines have to be connected (Figure 2e). To do so it is necessary to reposition the branch points in order to reflect the correct branching geometry (Figure 2f). In case of leakage, the front grows into the parenchyma, with its radius permanently increasing (Figure 2g). At some point the radius of the front exceeds a certain limit and the front is frozen (Figure 2h).

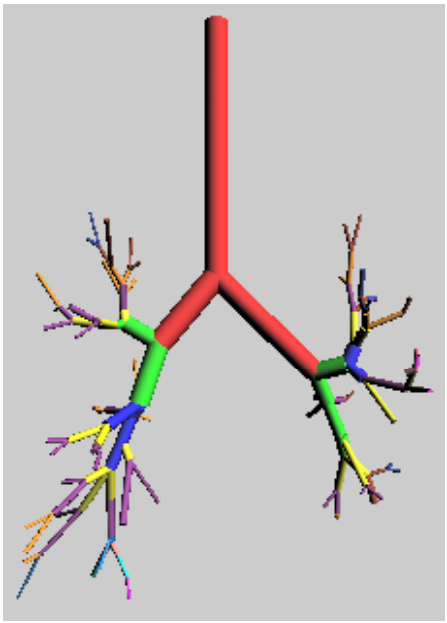


Figure 3: Model of the tracheobronchial tree.

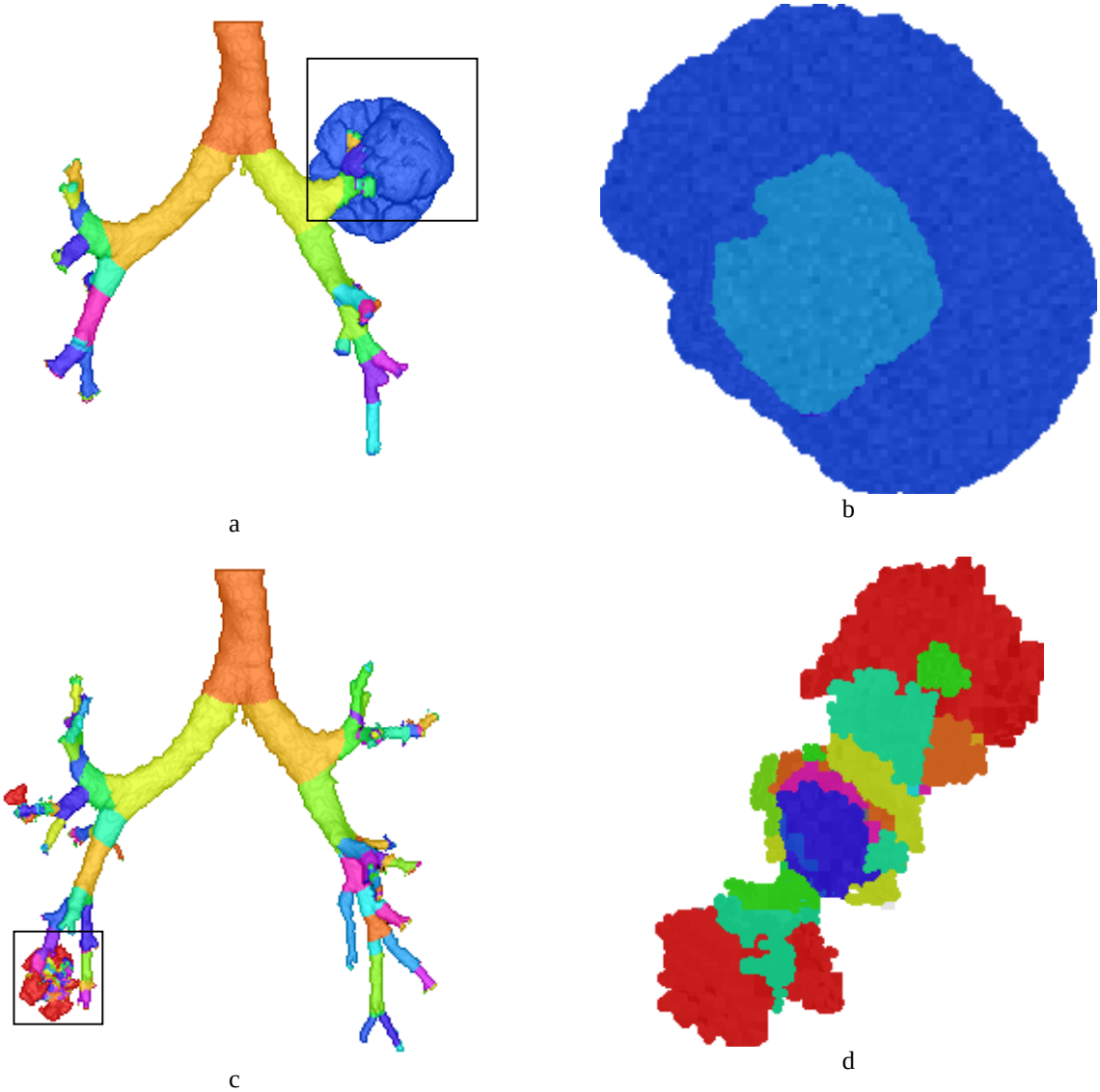


Figure 4: Results of leakage. The left side shows a surface rendering of the segmentation result with different gray values for different branches. The right side shows a cross sectional cut through the leaked region.; (a,b) Mushroom-like leakage into the parenchyma; (c,d) leakage into the parenchyma by creating lots of new branches.

Although the described process is able to create the skeleton of a tree like structure in a one-step process, it still runs the risk of leaking into the lung parenchyma. This is especially true due to the fact that the threshold value is chosen just above the bronchial lumen, i.e. close to the parenchyma. To address this problem we incorporate model information into the grow process. An abstract model of the tracheobronchial tree is shown in Figure 3. It is a model of connected cylinders of decreasing radius. The model which is presented in Figure 3 was generated from a real bronchial tree and thus reflects the correct bronchial anatomy. We make use of the fact that the radius of successive cylinders is decreasing and that each cylinder is connected to a very limited number of other cylinders (usually 3, the parent and two children). It is also possible to incorporate other information such as the correct bronchial anatomy, the branching angle, or the branching length.

Figure 4 shows two typical kinds of leakage. Different branches as detected by our algorithm are shown with different gray values. Figure 4a,b shows a mushroom-like leakage into the parenchyma. This kind of leakage usually occurs when the threshold value is chosen to completely include the parenchyma. Figure 4 c,d shows leakage that results in a large number of different branches. This kind typically occurs when the threshold value partially includes the parenchyma. Due to the noise within the parenchyma the algorithm detects a lot of branching. Both of these leakage types can be identified by our model information as follows.

In detail our segmentation algorithm works as follows (Figure 5). The main algorithm keeps a list of branches that have to be grown. This list is initialized with the trachea. The main part also keeps the *alive*, *trial* and *far* sets used within the fast marching algorithm[3]. After initialization the algorithm loops over a sequence of growing, branch detection, and branch validation:

Growing:
Consecutively, one branch is taken from the list and is grown according to the modified fast marching algorithm described above. Each branch keeps a reference to its initial radius (r_i) and compares this after every grow step to the actual radius. When the current branch approaches a bifurcation, the actual radius increases and finally exceeds the initial radius times a multiplication factor α (e.g. α =1.1).

Branch detection:
In case the actual radius exceeds $\alpha * r_{i,}$ we check for branching. Using α, we can greatly reduce the execution of the computationally expensive connectivity checking process. In case no branching is detected, α is increased about 0.1 and the grow process is continued. In the case of branching, the validity of the current branch is checked. This process is responsible for the detection of leakage.

Validation:
After branching occurred, the validity of the parent branch B of the branches B_i can be verified. Validation is the crucial point of the algorithm because it is responsible for rejecting branches that most probably represent leaked regions. This simple validation procedure will never be able to distinguish 100% of invalid branches from valid branches and can be further improved.

In our implementation of the validation procedure we use two criteria: radius and connectivity.

Radius: Since the grid point distribution of each branch is known from the segmentation result, it is possible to calculate its covariance matrix. Having in mind the cylindrical model, it is possible to estimate an average radius of the branch using the lowest two eigenvalues (EV) of the covariance matrix:

$$r_s = \frac{EV_2 + EV_3}{2}.$$

Since generally the radius is decreasing with increasing branch order a radius smaller than $\beta * r_{min}$ (with r_{min} being the smallest radius of all ancestors) indicates leakage. β is chosen to be greater than 1 to provide a safety margin to the internal variability of the radius of the branches. In general this criterion helps to detect the type of leakage shown in Figure 4a,b.

Connectivity: By checking the neighbor voxels of all surface voxels of a branch B, one can find the number of branches which are in the direct vicinity of B. This is an important validation criterion because experiments show that growing into the parenchyma often results in a large number of small branches growing in arbitrary directions. Thus an area with a large number of segments with no specified direction is created. This result is depicted in Figure 4c,d. If one compares the number of different branches in the direct neighborhood with the

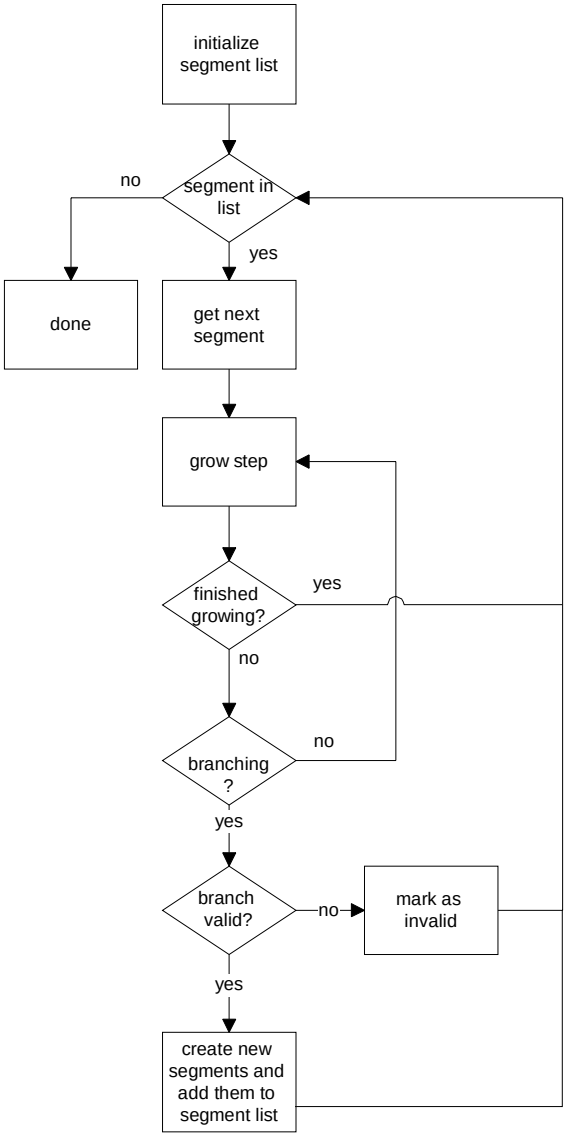


Figure 5: Flow chart of the bronchial segmentation algorithm

maximum number of allowed branches (γ) one can detect such a case of leakage. γ should be set to an integer number greater than three. Three neighbors is the usual case since a branch usually has a parent and two children. Three or more children are also possible, thus this parameter should be chosen carefully, not too low and not too large (e.g. γ=5).

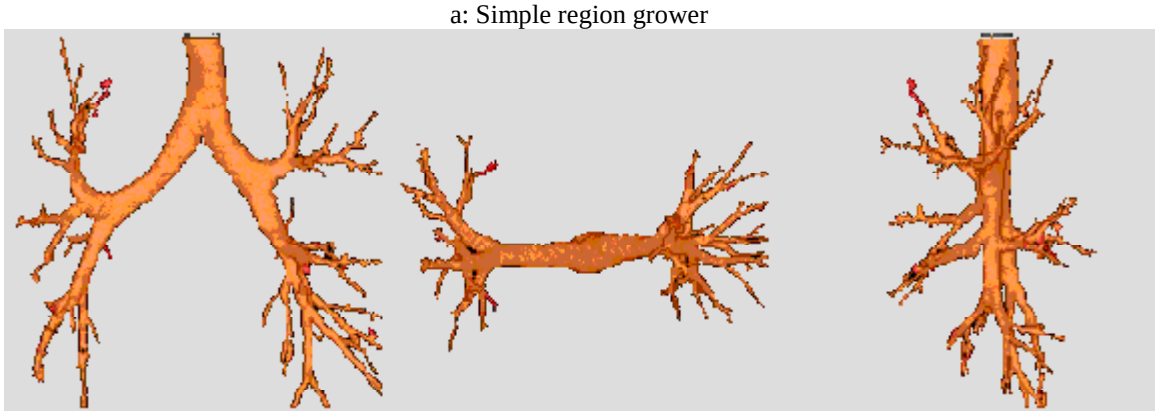
For valid branches, the unconnected regions of the front are used to initialize new branches which are stored in the branch list; for invalid branches they are discarded.

4. RESULTS

We applied two different kind of segmentation algorithms to lung data sets of 512^3 voxels with an isotropic resolution of 0.68mm. The first algorithm is a simple region growing that uses a threshold t as segmentation criterion. The second

method is the bronchial segmentation algorithm described in this paper. The segmentation results are depicted in Figure 6. It is obvious that the simple region growing algorithm can segment the bronchial tree already to a high degree, but at some point leakage occurs somewhere within the lingular bronchus. A detailed inspection of the data reveals that indeed valid parts are neglected in the segmentation process after leakage occurs. The algorithm is very fast and takes around 1s on a 800MHz PC. Our bronchi segmentation algorithm prevents the process from leaking into the parenchyma resulting in a better segmentation result for the whole bronchial tree. The algorithm takes less than 30s on an 800MHz PC and depends on only four parameters (t, α, β, γ) as described in the previous section. Preliminary studies show that the algorithm is able to segment 100% of the segmental bronchi (3^{rd}-order), 90% of 4^{th}-order sub-segmental bronchi, 60% of 5^{th}-order bronchi, and even approximately 25% of 6^{th}-order bronchi. In addition to the segmentation, it creates the skeleton of the bronchial tree avoiding invalid dead-end branches that frequently occur when conventional thinning algorithms are applied to create the skeleton. The skeleton shows deviations from the real bronchial tree only in the very distal parts in which the validation procedure rejected leaked regions. Up to now we did not apply any refinement algorithms to reduce the number of false branches in these distal regions. The detected centerline is very precise even for small segment radii. The skeleton of Figure 6b is shown in Figure 7. Therein each branch is depicted as a set of cylinders with the actual branch radius. In our application such a model has proven to be ideally suited for the navigation within the tracheobronchial tree.

a: Simple region grower



b: Bronchi segmentation algorithm

Figure 6: Results of bronchi segmentation using different segmentation algorithms. (a) A simple region growing results in leakage somewhere in the left upper lobe. The grow-process was manually stopped and further growing would rather increase the portion of the leaked region instead of growing valid regions. (b) Our bronchi segmentation algorithm is automatically terminated when no valid branches are available any more. Leakage occurs only to a very small extend because it is detected by the algorithm and then automatically stopped

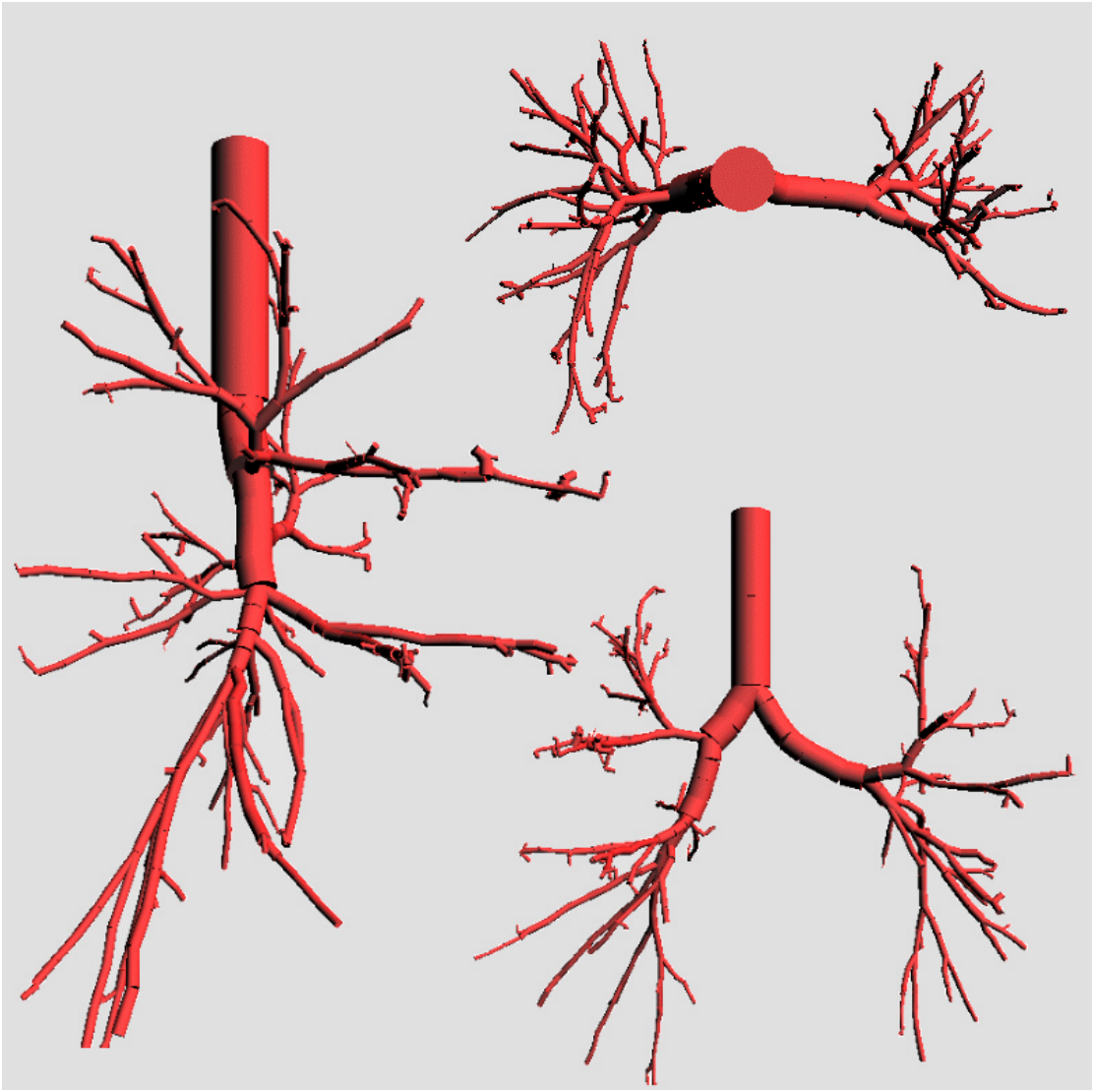


Figure 7: Cylinder representation of the bronchial skeleton. Each cylinder is assigned the radius of the actual branch.

5. CONCLUSION

In this paper we present a one-step segmentation/skeletonization algorithm that is capable of segmenting the tracheobronchial tree to a large extend. The results strongly depend on the resolution and the quality of the underlying data set. A detailed investigation of one test data set (512^3 voxel, 0.68mm isotropic resolution) reveals that 100% of the segmental 3rd-order bronchi, 90% of 4th-order sub-segmental bronchi, 60% of 5th-order bronchi and even approximately 25% of 6th-order bronchi could be segmented. Applying the algorithm to other data sets, we found that it is robust, although a detailed evaluation has not yet been performed. The skeleton is accurate even for small structures and only a very small number of false branches is generated in the very distal parts of the skeleton so that no pruning steps or refinements of the tree are necessary so far. The segmentation/skeletonization process takes less than 30s on an 800MHz PC.

REFERENCES

1. P. Rugalla, JT van Scheltinga, B. Hamm, Virtual Endoscopy and related 3D Techniques (Medical Radiology), Springer-Verlag Berlin Heidelberg, ISBN: 3540651578
2. W. E. Higgins, K. Ramaswamy, R.D. Swift, et al, Virtual Bronchoscopy for Three-dimensional Pulmonary Image Assessment: State of the Art and Future Needs, RadioGraphics 1998; 18: 761-778
3. T. Deschamps, L.D. Cohen , Minimal Paths in 3D images and application to virtual endoscopy, Lecture Notes in Computer Science: Computer Vision – ECCV 2000; 1843:543-557
4. A. Shahroknia, H. Soltanian-Zadeha, R. A. Zoroofia, Fast skeletonization algorithm for 3D elongated objects, Proc. of SPIE 2001; 4322:323-330
5. D. Selle and H. O. Peitgen, Analysis of the Morphology and Structure of Vessel Systems using Skeletonization, Proc. of SPIE 2001; 4321:271-281
6. D. Chen, B. Li, Z. Liang, et al, A tree-branch searching, multiresolution approach to skeletonization for virtual endoscopy, Proc. of SPIE 2000; 3979:726-734
7. T.Y. Law, P.A. Heng, Automated extraction of bronchus from 3D CT images of lung based on genetic algorithm and 3D region growing, Proc. of SPIE Vol. 3979:906-916
8. R.M. Summers, D.H. Feng , S.M. Holland, et al. Virtual Bronchoscopy: Segmentation method for real-time display, Radiology 1996; 200:857-862
9. M. Sonka, W. Park, E.A. Hoffmann, Rule based detection of intrathoracic airway trees, IEEE Trans on Med Img 1996; 15(3):314-326
10. J.A. Sethian, Level Set Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Material Science, Cambridge University Press 1999
11. R.M. Summers, M.C. Sneller, C.A. Langford, et al. Improved virtual bronchoscopy using a multi-slice helical CT scanner, Proc. of SPIE 2000, Vol. 3978:117-120