

# Automatic Centerline Extraction for 3D Virtual Bronchoscopy

Tsui Ying Law and Pheng Ann Heng

Department of Computer Science and Engineering  
The Chinese University of Hong Kong,  
Shatin, N.T., Hong Kong  
{tylaw, pheng}@cse.cuhk.edu.hk

**Abstract.** Centerline extraction is the basis to understand three dimensional structure of the lung. In this paper, an automatic centerline determination algorithm for three dimensional virtual bronchoscopy CT image is presented. This algorithm has two main components. They are *end points retrieval algorithm* and *graph based centerline algorithm*. The end points retrieval algorithm first constructs a binary tree which links up all necessary center points of each region in each slice from segmented lung airway tree volume data. Next, it extracts the end points of the lung airway tips from the binary tree. The graph based centerline algorithm reads the end points and applies distance transform to yield a distance map which shows all shortest paths from the start point to those end points. Then, modified Dijkstra shortest path algorithm is applied in the centerline algorithm to yield the centerline of the bronchus. Our algorithm is tested with various CT image data and its performance is encouraging.

## 1 Introduction

The idea of using virtual reality is currently of great interest to the medical imaging community. Virtual reality system can provide an initial assessment of the condition of patients. Bronchoscopy is a medical diagnosis for evaluating the endobronchial anatomy. Virtual reality and volumetric imaging develop so rapidly that their technologies are matured enough to create a simulated (virtual) environment for medical surgery, which takes less cost and risks. Virtual bronchoscopy [7] is a new concept to combine volumetric imaging and virtual reality technology for simulating bronchoscopy. It is gaining public attention because of its potential for decreasing discomfort and inconvenience, considerably lower cost and risks, in comparison with routine bronchoscopic screening procedures. Simulation technology makes it possible for an user to experience adverse scenarios without risk to human life or damage to expensive equipment [2]. In addition, it is useful for training medical students or physicians to achieve better surgery skills. Comparing the virtual and real surgery, although it has the physical biopsy, color/texture limitations, virtual one takes the advantages of measurements providing, reproducibility, flexibility and non-invasiveness.

We are currently developing a virtual bronchoscopy system for training and diagnosis purposes. The system allows healthcare providers to practise procedures in an environment where mistakes do not have serious consequences. In addition, it lowers risk associated with training on human patients, avoids the use of animals for training, and establishes standards and optimization of specific procedures.

One required function of the virtual bronchoscopy system is to extract and recognize the 3D structure of bronchus in lung area. The automated extraction algorithm of the bronchus area [5] is an enhancement of algorithm described in [8] and [9]. It is for the recognition of bronchus in three-dimensional CT images, which is based on region growing method. Region growing method [11] is the most popular and widely used method for the detection of tree structured objects such as bronchus and blood vessels [4]. The algorithm is applied to real 3D CT images and experimental results have been given to demonstrate its ability to extract bronchus area and the result is satisfactory.

We can extract the bronchus. However, they are just a simple set of voxels. To understand their 3D structure, additional analysis is necessary. Centerline extraction [4] is the basis to understand 3D structure. To extract centerlines, thinning processing is usually applied. However, it produces skeletons which may not reflect the true shape of the original pattern. Therefore, a more accurate algorithm for extracting the centerline of bronchus is desired.

The organization of this paper is given as follows. After giving an introduction in Section 1, the detailed description and experimental results of our centerline algorithm are presented in Sections 2 and 3 respectively. Conclusion is given in Section 4.

## 2 Automatic Centerline Determination Algorithm

Our automatic centerline determination algorithm is composed of two main components. The first component is *end points retrieval algorithm* which converts segmented lung airway tree volume data into a set of end points. The second component is *graph based centerline algorithm*. The algorithm read the end points and it yields a distance map which shows all shortest paths from the start point to those end points. Those end points can be used to construct a set of centerlines of the bronchus.

### 2.1 End Points Retrieval

Assume that the slices of the volume data are in axial format, we first determine all the center points of every region in each slice. In each slice, every pixel is scanned once to perform region growing searching. Segmented volume data can be divided into two categories: feature or non-feature pixel. The feature pixels represent the lung tree airway extracted, while the non-feature pixels represent those remaining areas. When a feature pixel is being visited, region growing is started from the pixel and search its eight neighbors recursively. Pixels are

marked when they have been visited. Those unmarked feature pixels will start region growing to form other regions. After the whole slice is scanned, a few regions should be extracted. The center of mass of each region is determined and it is treated as the center point of that region.

After the center points of all slices are determined, they will be linked up by our center points linking algorithm. Based on the bifurcation characteristics of the lung tree volume, we link up those center points starting from the main bronchus to tiny airway tips. By the algorithm in [5] which adopts genetics algorithm to determine the start point, a seed point which belongs to the main bronchus is generated. The center point of the corresponding region that contains the seed point is then used as the initial point in the center points linking algorithm.

The initial point is inserted into a linked list which will contain subsequent center points for processing. Let  $L$  be the linked list,  $p$  be the first element of the list and  $\{c_1, c_2, \dots, c_n\}$  be the set of unlinked center points which are in the previous or next slices of  $p$ . Distance between  $p$  and all elements in  $\{c_1, c_2, \dots, c_n\}$  will be calculated so that any point will be linked up to  $p$  if it is close to  $p$ .

To measure the closeness between two points, there are two threshold values *distance\_threshold* and *possible\_distance\_threshold* used in our algorithm. Let  $x_l$  and  $x_{l+1}$  be two center points in  $l^{th}$  and  $l+1^{th}$  slices respectively. If the distance between  $x_l$  and  $x_{l+1}$  is not larger than *distance\_threshold* and  $x_l$  has less than 2 children,  $x_l$  is linked to  $x_{l+1}$  which represents that  $x_l$  is the parent of  $x_{l+1}$ . If the distance between  $x_l$  and  $x_{l+1}$  is larger than *possible\_distance\_threshold*,  $x_l$  and  $x_{l+1}$  do not have child-parent relationship. If the distance falls between *distance\_threshold* and *possible\_distance\_threshold*, they may have parent-children relationship and further test is needed. In this case,  $x_l$  is inserted into the list  $L$  so that all its preceding points in the list have the same slice number but with smaller minimum distance. When  $x_l$  is extracted again and it has one child or less,  $x_l$  is linked to  $x_{l+1}$  if  $x_{l+1}$  has not yet been selected from other points. Then  $x_l$  is the parent of  $x_{l+1}$ . If  $x_{l+1}$  is a child point of  $x_l$ , it is then inserted into the end of  $L$  so that its children will be found later. Every point should have only one parent. Once it is selected from a point, the child-parent relationship is established and no other point can be its parent therefore.

The center points linking algorithm is shown in Fig. 1. After the algorithm is completed, a binary tree which links up all necessary center points is yielded. In the binary tree, end points are the points which do not have any child. It implies that they are corresponded to the lung airway tree tiny tips. Since the binary tree is created, those end points can be easily extracted by the traversal of the tree.

## 2.2 Graph Based Centerline Algorithm

Based on chamfer distance transform and Dijkstra's single source shortest path algorithm [10] [3], Blezek [1] applies a centerline algorithm to virtual endoscopy. However, he mainly concerns one-start-point-one-end-point endoscopy such as colonoscopy and user is required to supply the start and end points. His algorithm

```

LinkCenterPoint(list  $L$ ){
1.  While list  $L$  not empty{
2.      Extract the first member  $p$  of the list  $L$ 
3.      Locate all unlinked center points  $\{c_1, c_2, \dots, c_n\}$  in the previous and
        next slices of  $p$ 
4.      Calculate the distance from  $\{c_1, c_2, \dots, c_n\}$  to  $p$  as  $[d_1, d_2, \dots, d_n]$ 
5.      Sort the center points  $[c_1, c_2, \dots, c_n]$  by the corresponding  $[d_1, d_2, \dots, d_n]$ 
6.      if  $d_1 \leq \text{distance\_threshold}$ 
7.          link up  $p$  to  $c_1$ 
8.          insert  $c_1$  into list  $L$ 
9.      if  $d_2 \leq \text{distance\_threshold}$ 
10.         link up  $p$  to  $c_2$ 
11.         insert  $c_2$  into list  $L$ 
12.      if  $d_1$  or  $d_2 > \text{distance\_threshold}$  and  $\leq \text{possible\_distance\_threshold}$ 
13.         if  $p$  not yet appear in the linked list  $L$ 
14.             insert  $p$  into the list  $L$  in a suitable position
15.         else link up  $p$  to the corresponding point ( $c_1$  or  $c_2$ )
        }
    }
}

```

**Fig. 1.** Center Points Linking Algorithm

cannot be applied directly to virtual bronchoscopy, which has many branches. In order to develop an automatic algorithm for centerline extraction for virtual bronchoscopy, we introduce the following modifications.

Our centerline algorithm first calculates a three dimensional chamfer 3-4-5 distance map of the volume data which is a two-pass procedure. All voxels in the interior of the object are labelled with the distance to the nearest background voxel. In other word, a voxel in the center interior of the object should have a larger distance value as it is farther apart from the background comparing with those boundary voxels. From the distance map, the maximum distance value  $V_{Max}$  is determined. All voxel distance values are squared after subtracted from  $V_{Max}$  to achieve the result that more interior voxels have smaller distance values and closer to boundary voxels have larger distance values relatively. Thus, voxel has zero distance value is as far from the boundaries of the object as possible. The squaring is performed to encourage the path to maintain a medial path.

A weighted directed graph  $G = (V, E)$ , where  $V$  is the set of vertices and  $E$  is the set of edges in the graph is constructed. All voxels are considered as nodes (vertice). Edges are created between a vertex and its 26-connected neighbour. The weight of a edge is assigned as the distance value of the voxel in in-direction. For example if the distance value of voxel  $V_1$  is four, then the weight of edges  $(V_1, V_0)$  and  $(V_1, V_2)$  are four.

A general method solving the single-source shortest-path problem is known as Dijkstra's algorithm. It is a prime example of greedy algorithms. Greedy algorithms generally solve problems in stages by doing what appears to be the best thing at each stage. Similarly Dijkstra's algorithm proceeds in stages. In

```

dijkstra( $G$ ,  $StartPt$ ){
1.  initial  $distance[i] = \infty$  and previous vertices are  $NULL$ 
2.  set  $distance[StartPt] = 0$ 
3.  build a binary minimum heap which is keyed to  $distance[]$ 
4.  while heap is not empty {
5.      retrieve the minimum item  $v$  from the heap
6.      for each adjacent  $w$  to  $v$ 
7.          if  $w$  not out of bound and  $w$  is a feature point
8.              update  $w$  distance and previous vertex if required
9.              update heap if  $w$  information is modified
    }
}

```

**Fig. 2.** Modified Dijkstra Shortest Path Algorithm

each stage, Dijkstra's algorithm selects a vertex  $v$  which has the smallest distance among all the unknown vertices, and declares that the shortest path from source to the vertex  $v$  is known. The remainder of a stage consists of updating the values of the neighbours of the vertex  $v$ . A priority queue  $Q$  that contains all the unknown vertices keyed by their distance values is maintained. It is practical to implement the priority queue  $Q$  with a binary heap [10] [3]. The resulting algorithm is sometimes called the modified Dijkstra algorithm. Binary heaps have two properties, namely, a structure property and a heap-order property. A heap is a binary tree that is completely filled, with the possible exception of the bottom level, which is filled from left to right. Such a tree is known as a complete tree. It can be represented in an array and no links are necessary. For any element in array position  $i$ , the left child is at position  $2i$ , the right child is in the cell after the left child ( $2i + 1$ ), and the parent is at position  $i/2$ . Thus not only are links not required, but the operations required to traverse the tree are extremely simple and can be very fast on most computers. The heap-order property allows operations to be performed quickly. In a heap, for every node  $X$ , the key in the parent of  $X$  is smaller than or equal to the key in  $X$ , with the exception of the root which has no parent. By the heap-order property, the minimum element can always be found at the root. Thus, we can retrieve the minimum element quickly.

When the Dijkstra algorithm as shown in Fig. 2 is completed, the previous vertex in the shortest path from the source vertex  $StartPt$  to each vertex is determined. The previous vertex should be a 26-neighbour of the current vertex. Therefore, three dimensional 26-connected chains are generated from the source vertex  $StartPt$  to each feature vertex. To improve the efficiency of the procedure, only those feature vertices are considered. If the vertices are non-feature (background) one, their previous vertex should be remained  $NULL$  which is not changed since initialization.

The centerline required is constructed by extracting the shortest path from each end point to the source point. It is accomplished by following the chain code

from end points back to source point. The voxels visited are marked. Combining all the visited voxels will form the centerline of the object.

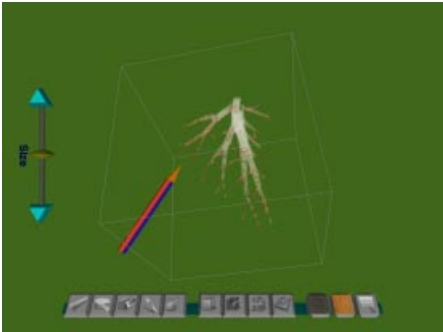
### 3 Experiment and Discussion

The automatic centerline extraction algorithm is implemented and tested on real 3D CT data which is obtained from a dog lung. Seven sets of lung airway tree segmented data are generated by the method mentioned in [5]. Among the seven image sets a reduced resolution image set that is composed of 64 images, where each of them is a 128x128 8-bit gray-scale image, are prepared for comparison. Other six image sets contain around 185 images where each of them is a 256x256 8-bit gray-scale image. None of pre-processing operations such as smoothing, interpolation is applied. All experiments are performed on Sun UltraSparc 5/270 workstation with Solaris 2.6 OS, 512MB RAM and 100Mbps network speed. Table 1 presents the running time of the algorithm. The calculation time is approximately linear with number of object voxels.

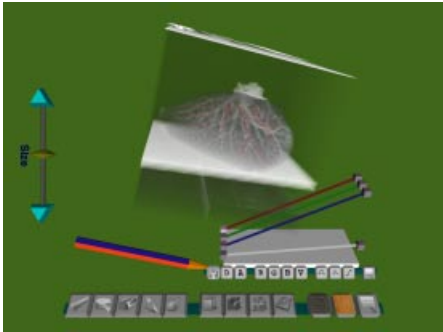
Data	Volume Size(Voxels)	Object Voxels	End Point Time(s)	DT Time(s)	Dijkstra Time(s)	End Point Voxels	Center -line Voxels
set1	256x256x185	54398	4.3	5.79	3.85	91	1490
set2	128x128x64	9657	0.39	0.53	0.43	47	465
set3	256x256x185	60025	4.32	5.7	3.69	93	1356
set4	256x256x180	72378	4.3	5.7	4.42	129	1784
set5	256x256x185	98885	4.43	5.99	4.95	162	1664
set6	256x256x180	91697	4.42	5.98	5.36	206	2565
set7	256x256x200	99094	4.88	6.61	5.78	98	1466

**Table 1.** CPU Running Time of End Points Retrieval Algorithm, Distance Transform and Dijkstra Shortest Path Algorithm

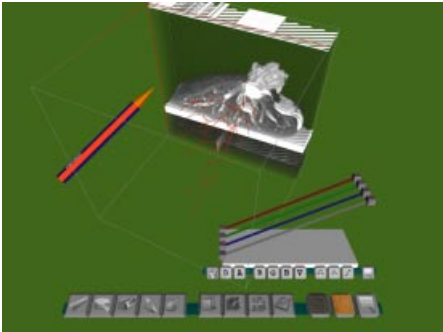
*distance\_threshold* and *possible\_distance\_threshold* are set to be 3 and 26 respectively in the end points retrieval algorithm. User can alter the threshold whenever necessary. For a 12MB lung airway tree volume data, it spends about 4 seconds to generate over 90 end points. The graph based centerline algorithm takes about 10 seconds to complete, and one third of the execution time is contributed to the completion of the modified Dijkstra shortest path algorithm. One of the results in Blezek [1] which consists of 99054 object voxels with a 256x256x256 volume size requires 20 seconds to complete. The algorithm in Blezek [1] requires user to input a start point and an end point to his shortest path searching algorithm. On the contrary, our algorithm includes end points searching function that does not require user to input the start point. In the case of similar object voxel number, our algorithm requires 12 seconds including end points searching, distance transform and Dijkstra shortest path searching. Our algorithm is much more efficient than Blezek's algorithm.



(a) Transparent Lung Tree Airway Data and Centerline Generated



(b) Transparent Lung Volume Data and Centerline Generated



(c) Visualization of Lung Volume in Cutbox View



(d) Visualization of Segmented Lung Tree in Cutbox View

**Fig. 3.** Various Visualization Results

The extracted centerline is visualized with 3D texture mapping techniques [6]. This method is a direct data visualization technique that is similar to ray casting. Three dimensional textures are a logical extension of 2D textures. In 3D textures, texels become unit cubes in texel space. The 3D texture is used as a voxel cache, processing 2D layer each time by all rays simultaneously. Fig. 3 show the images of the reconstructed centerlines. The centerlines are shown to be extracted satisfactorily.

## 4 Conclusion

An automatic centerline determination algorithm from CT images for three dimensional virtual bronchoscopy is presented. The end points retrieval algorithm extracts end points of the lung airway tips. Distance transform and modified Dijkstra shortest path algorithm are then applied in the centerline algorithm which yields the centerline of the bronchoscopy. Experimental results show that our algorithm is more efficient than algorithm in Blezek [1] which is the basis of our centerline algorithm. Our test cases include various CT image data sets. For a typical 256x256x180 segmented lung tree airway volume data, it requires about 12 seconds for the whole centerline determination procedure. The experimental results show that the performance of our algorithm is encouraging.

## Acknowledgments

This work is supported by Hong Kong RGC Earmarked Research Grants CUHK 4167/97E.

## References

- [1] Daniel J. Blezek, Richard A. Robb, *Centerline Algorithm for Virtual Endoscopy based on Chamfer Distance Transform and Dijkstra's Single Source Shortest Path Algorithm*, Part of the SPIE Conference on Physiology and Function from Multi-dimensional Images, San Diego, California, SPIE Vol. 3660, Feb 1999.
- [2] Richard D. Bucholz, MD, FACS, *Advances in Computer Aided Surgery*, CAR'98.
- [3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, *Introduction to Algorithms*, The MIT Press, McGraw-Hill Book Company, 1990.
- [4] Fumikazu Iseki, Tsagaan Baigalmaa et al., *Extraction of 3D Tree Structure of Blood Vessels in Lung Area from Chest CT Images*, CAR'98, 1998.
- [5] Tsui-Ying Law, Pheng-Ann Heng, *Automated Extraction of Bronchus from 3D CT Images of Lung*, pp. 906-916, Medical Imaging 2000: Image Processing, Proceedings of SPIE Vol. 3979(2000), 2000.
- [6] T. McReynolds et al., *Advanced Graphics Programming Techniques Using OpenGL*, SIGGRAPH 98 Course Notes 17, July 1998.
- [7] J. R. Mellor, G. Z. Yang et al., *Virtual Bronchoscopy*, Image Processing and its Applications, 4-6 July 1995, Conference Publication No. 410, IEE, 1995.
- [8] Kensaki Mori, Jun-ichi Hasagawa et al., *Automated Extraction and Visualization of Bronchus from 3D CT Images of Lung*, Lecture Notes in Computer Science 905, Computer Vision, Virtual Reality and Robotics in Medicine, First International Conference, CVRMed'95, Nice, France, Springer, pp. 542-548, April 1995.
- [9] Kensaku Mori, Jun-ichi Hasegawa et al., *Recognition of Bronchus in Three-Dimensional X-ray CT Images with Applications to Virtualized Bronchoscopy System*, Proceeding of ICPR'96, IEEE, 1996.
- [10] Mark Allen Weiss, *Data Structures & Algorithm Analysis in C++*, Addison Wesley Books, Second Edition, 1999.
- [11] Terry Yoo, *Segmentation and Classification*, Siggraph 93 Course Notes 21, pp. 12-15, Aug. 1993.