# Automatic Extraction of Bronchus and Centerline Determination from CT Images for Three Dimensional Virtual Bronchoscopy

Law Tsui Ying

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Computer Science & Engineering

Supervised by:
Dr. Heng Pheng-ann


June 2000

論文題目：配合三維虛擬肺部內窺檢驗系統及電腦素描圖像的自動化抽取肺部氣管和確定居中線算法

作者　　：羅翠瑩
修讀學位：哲學碩士
學部　　：計算機科學及工程
學院　　：香港中文大學
日期　　：二零零零年六月

摘要

在本論文中，我們提出了兩項算法來配合三維虛擬肺部內窺檢驗系統。它們分別是設計及應用在電腦素描圖像數據上的自動化抽取樹形結構肺部氣管算法和自動化居中線確定算法。

在自動化抽取樹形結構肺部氣管算法中，我們利用了三維種子區域生長算法來協助獲得分割後的肺氣管範圍。首先基因算法根據氣管樹的幾何特點，如形狀、位置和長度來取得一個種子點。另外再以肺部和肺部氣管的體積比例，產生出最佳的定限值。用這最佳的定限值，氣管區域被抽取出，並被重組和用三維紋理映射方法來立體展現它的內部結構。

然後，運用我們的自動化居中線確定算法在這些已被抽取的肺氣管圖像數據上。這項算法有兩個主要組合。它們是末端點檢索算法和以圖形為基礎的居中線算法。末端點檢索算法取得肺氣管的末端點資料後，再利用最短的路徑算法和距離變換來獲得肺氣管的居中線。我們用真實的電腦素描圖像數據來做不同的實驗，而實驗結果是令我們鼓舞的。

Abstract of thesis entitled: Automatic Extraction of Bronchus and Centerline Determination from CT Images for Three Dimensional Virtual Bronchoscopy

Submitted by: Law Tsui Ying
for the degree of Master of Philosophy
at The Chinese University of Hong Kong in June 2000

## Abstract

In this paper, we propose (*i*) a method to automate the segmentation of airway tree structures in lung from a stack of grey-scale computed tomography (CT) images; and (*ii*) an automatic centerline determination algorithm for three dimensional virtual bronchoscopy CT image. In the automated segmentation algorithm, a three-dimensional seeded region growing is performed on images without any preprocessing operation which may assist to obtain the segmented bronchus area. We first use genetic algorithm (GA) to retrieve a seed point and the algorithm is based on the geometric features (shape, location and size) of the airway tree. By the feature of the size of the lung and airway tree, an optimal threshold value is yielded. The final extracted bronchus area with the optimal threshold value is reconstructed and visualized by three dimensional texture mapping method. Then, the algorithm is followed by our centerline extraction algorithm. The extraction algorithm has two main components, *end points retrieval algorithm* and *graph based centerline algorithm*. The end points retrieval algorithm extracts end points of the lung airway tips. Distance transform and the modified Dijkstra's shortest path algorithm are then applied in the centerline algorithm which yields the centerline of the bronchus. Our centerline extraction algorithm is tested with various real CT image data and phantom data. The inspiring experimental results show that our algorithm has good performance.

# Acknowledgments

I would like to thank Professor Heng Pheng-ann for his perceptive comments and advice.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

There are four common imaging modalities [60]: X-ray computed tomography (CT), nuclear medicine imaging including both positron emission tomography (PET) and single photon emission computed tomography (SPECT), magnetic resonance imaging (MRI) or nuclear magnetic resonance image (NMR), and ultrasonography or ultrasound. Some methodologies are more appropriate for specific data created by the modality. To produce optimal visualization, understanding the strength and weakness of those mechanisms is necessary.

Traditionally, all images are viewed on film including the volume data images due to CRT displays' fundamental resolution limitations. Nowadays, there are some new directions for virtual reality in medicine. Some medical applications attempt to merge technology into immersive display (virtual reality). Robb [45] mentions that, by using virtual reality, users can manipulate the virtual object similar to that of real objects. The viewers can interactively visualize the illustrated objects. Furthermore, he suggests that engaging other senses such as touch, hearing or even smell can enrich the visualization.

Recently, the idea of using virtual reality is 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 develops

so rapidly that their technology is matured enough to create a simulated (virtual) environment for medical surgery, which takes less cost and risks. Virtual bronchoscopy [33] [14] is a new concept to combine volumetric imaging and virtual reality technology. As its name tells, the concept is applied to 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 navigation to experience adverse scenarios without risk to human life or damage to expensive equipment [6]. In addition, it is useful for training medical students or physicians to achieve better surgery skills. Comparing the virtual and real surgery [1], although it has the physical biopsy, color/texture limitations, virtual one takes the advantages of measurements providing, reproducibility, flexibility and non-invasiveness.

Since the research potential of developing virtual bronchoscopy system is very high and the advantages of using such system are prominent, we plan to develop a virtual bronchoscopy system to train medical students. Moreover, it 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.

In order to develop the virtual bronchoscopy system, it is necessary to extract and recognize the three dimensional structure of bronchus in lung area. The automated extraction algorithm of the bronchus area [26] is an enhancement of algorithm described in [34] and [35]. It is for the recognition of bronchus in three-dimensional CT images, which is based on region growing method. Region growing method [59] is the most popular and widely used method for the detection of tree structured objects such as bronchus and blood vessels [22]. The algorithm is applied to real three dimensional 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 three dimensional structure, additional analysis is necessary. Centerline extraction [22] is the basis to understand three dimensional 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. Our centerline extraction algorithm uses distance transformation [4] [3] approach to label each voxel in the object with a medialness messure. Dijkstra's single-source shortest path algorithm which is a graph based technique to yield the centerline precisely. Our centerline extraction algorithm is based on Blezek [2] method which he applies to virtual endoscopy system. Endoscoopy system is for examining visually the interior of a bodily canal or a hollow organ such as the colon, bladder, or stomach. Bronchoscopy system is for inspection of the interior of the bronchi. Bronchoscopy system and endoscopy system differ in the applied object. The latter one usually is applied on hollow organs and cavities without special consideration of the branching feature. In virtual bronchoscopy system, careful consideration of the branching characteristic is required. To understand more about the bronchus features, the introduction to the structure of bronchus is provided.

## 1.1 Structure of Bronchus

The lung [10] contains a complex system of branching trees that conduct air and blood down to the small gas exchanging regions [42]. Each lobe of a lung has an airway tree which contains the air we breath, and vascular trees which contain blood. An blood filled airway wall surrounds throughoutly the airway tree [41]. The bronchus is composed of a pipe structure and contains air in it. It starts from the end of trachea and branches repeatedly like a tree while extending in the lung [34]. Each branch divides into two smaller branches consistently. The two daughter branches from the same parent often differ in diameter and/or

in length. The dichotomy is hence irregular. However, as a rule, the diameter of the daughter branches is smaller than that of the parent branch. Therefore, the diameter and the length of the airways become progressively smaller as one proceeds from the trachea to the periphery [56]. The airway wall is very thin and its inner area is filled with air. If it runs across the slice vertically which are axial images from top to bottom of body, its cross section on a slice is observed as an approximately circular ring. However, its border is often vague and hardly recognized even for human eyes [35]. Using computed tomography (CT), we can obtain multislice lung data sets.

## 1.2 Existing Systems

The advantages and disadvantages of several existing systems are discussed. Virtual Endoscope System, Virtual Reality Surgical Simulator, Automated Virtual Colonoscopy and QUICKSEE are to be introduced.

### 1.2.1 Virtual Endoscope System (VES)

VES [21] is a system with force sensation to train inexperienced young doctors to simulate operations that require special technical skills, i.e. endoscopic insertion. The force simulation mechanism is developed with the use of four rubber rollers and differential gears. This simple structure is easy to control and stable for the linear and rotational drive of the endoscope.

### 1.2.2 Virtual Reality Surgical Simulator

Virtual Reality Surgical Simulator [39] is an interactive system for the training and assessment of suturing technique in the context of end-to-end anastomosis. It is composed by surgical tools with force feedback, a three dimensional visual display of the simulated surgical field, physics-based computer simulations of

the tissues and tools, and software to measure and evaluate the trainee's performance. The main focuses of this system are the quantification and evaluation of a user's performance and the realistic sense of the touch.

### 1.2.3 Automated Virtual Colonoscopy (AVC)

AVC [20] is a computer-aided diagnostic system for automatically analyzes the three dimensional images of the colon and highlights areas of potential lesions by color-enhancing in order to focus a physician's attention to suspicious areas. Human visual analysis is time-consuming, tedious, and often prone to error of interpretation. It can measure the colon's wall thickness and provide the shape analysis. However, it has the false positive findings problem.

### 1.2.4 QUICKSEE

It is a digital endoscopic system, QUICKSEE [43] [18], to achieve the following tasks: (i) dynamic interactive visualization of anatomical structures that exist in a high-resolution three dimensional image, and (ii) extensive quantitative analysis of these structures. The proposed system enables noninvasive and unrestricted exploration of the virtual anatomy and can be useful as a diagnostic tool or as a surgical (endoscopy) planning aid. The system is particularly useful for analyzing complex three dimensional branching structures, such as the coronary arteries and pulmonary airways and arteries. Usually systems either lack the means for: (1) interactive exploration along arbitrary paths and analysis of complex (branching) structures; or (2) do not permit endoscopic exploration interactively on real image data. QUICKSEE fills these requirments. However, it does not use any virtual reality technology. The user would not have the sense of immersion.

## 1.3 Organization of Thesis

The organization of this paper is given as follows. After giving an introduction which includes the introduction of structure of bronchus and existing systems in Chapter 1, we describe three dimensional visualization in medicine, several advanced segmentation methods and overview of centerline determination methods in Chapters 2, 3, 4 respectively. Detailed description of our automated extraction algorithm and centerline algorithm are presented in Chapters 5 and 6. Experimental results are presented in Chapter 7. Conclusion is given in chapter 8.

# Chapter 2

# Three Dimensional Visualization in Medicine

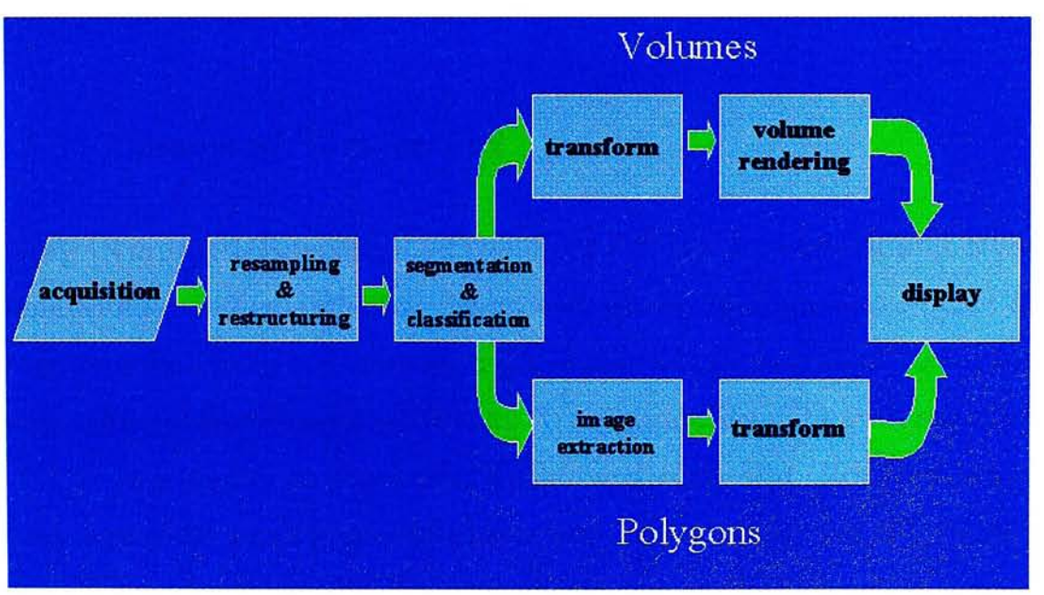



**Figure 2.1**: The medical visualization pipeline from acquisition to display

Visualization is for displaying the data to user in a convenience way for having further visual details from the data. It should provide rapid and accurate analysis. User can always cross reference the original data to the visualization. There are typical steps for the three dimensional visualization in medicine [60] as shown in Figure 2.1. It is composed by a series of complex steps. They in-

clude acquisition, restructuring, classification, rendering, and display in order.

## 2.1 Acquisition

X-ray computed tomography (CT), nuclear medicine imaging including both positron emission tomography (PET) and single photon emission computed tomography (SPECT), magnetic resonance imaging (MRI) or nuclear magnetic resonance image (NMR), and ultrasonography or ultrasound are four common imaging modalities. It is essential to understand the mechanisms' strengths and weakness to produce optimal visualization. Some mechanisms are designed for particular data created by the modality. Using spiral computed tomography (S-CT) enables volumetric data acquisition during a very short time spans [53]. Our system is specially applied on CT data.

### 2.1.1 Computed Tomography

X-ray Computed Tomography [Yoo98] or CT is the most familiar form of three dimensional medical imaging. To protect the technologists and other clinic staff from exposure from routine use, a CT scanner is a room sized X-ray instrument with a shielded environment It is an X-ray modality. The process of generating a CT scan is similar to generating a standard x-ray film. However, while a single x-ray exposure generates a complete film based exam, the CT image must be reconstructed from multiple views. The advantage of being an x-ray modality is that laymen and clinicians alike have considerable intuition when dealing with x-ray based images. The concepts of dense objects like bone absorbing more photons relative to less dense tissues like muscle or fat come naturally from our experience and expectations about x-ray imaging.

A typical CT scanner can generally acquire the data for an transaxial slice in a matter of seconds (within 1 to 5 seconds). An exam can include several series of slices, in some cases with and without pharmaceutical contrast agents

injected into the patient to aid in diagnostic reading. Slices can be spaced such that they are either overlapping or contiguous, though some protocols call for gaps between the slices. A large study can include well over 100 separate 512 by 512 pixel images. The radiation dose from a CT scan is comparable with that of a series of traditional X-rays.

## 2.2 Resampling

The data usually is not in standard format after the acquisition process. It is necessary to resample and restructure the data to fullfil the limited flexibility of visualization packages. This process is lack of generality. It is specific to the visualize software and the particular data sets.

## 2.3 Segmentation and Classification

The division of an image into coherent regions along some syntactic (local image characteristic) criteria is segmentation. Classification is used for the labeling of those regions with the aid of a user. The simplest classifier mechanism is intensity windows (multiple thresholding classification) which are mainly useful in dealing with X-ray CT data.

Segmentation is the process of recognizing objects within an image. The image will be divided into a number of disjoint areas which have different features. To specify a single object of an image, the first task is to determine for every pixel of the image whether it represents or belongs to the interested object or not. Generally, there are no general segmentation method which can recognize any object in any kind of image. The segmentation method used depends the characteristics of the objects and the given image.

Basic segmentation techniques can be categorized into three classes, feature thresholding methods, region based methods, and edge oriented methods.

### 2.3.1 Segmentation by Thresholding

Segmentation by thresholding assigns all pixels with an intensity at or above a fixed value (threshold) to an object. The object is represented by those pixels with intensity values higher than the threshold. It is simple and fast. However, it is relied on threshlod determination and it works only on a very limited number of images as it considers only the intensity value without any consideration of spatial location. Another basic limitation of thresholding is that it cannot classify image elements with multiple objects.

The other two classes methods use information from pixels in an arbitrarily large neighborhood around a pixel. They assume that objects in the image represent a closed region. There are two general approaches, finding the border of the region (i.e. the edge oriented methods) or finding the interior of the region (i.e. the region based methods including texture analysis and region growing).

### 2.3.2 Segmentation by Texture Analysis

For the texture analysis, the granularity is dependent on the resolution at which the texture is observed and medical images appear to be simple ones. A inherent problem is that the uncertainty of the location of the local texure. If given a small neighborhood, the location is relative precise, but the texture measure is uncertain. On the other hand, if a large neighborhood is given, the texture is measured with high accuracy, but the location is uncertain. Therefore, a local texture segmentation will never show the precise border lines of a region which need some refinement to be done.

### 2.3.3 Segmentation by Region Growing

Segmentation by region growing requires a pixel or a set of pixels (region) which belongs to the desired object. Then its neighborhood is examined iteratively to decide whether the current examining pixel belongs to the interested object or

not. This method has the advantage that it is capable of correctly segmenting spatially separated objects although they have the same features and it can generate connected regions. However, it is more computationally expensive than single pixel techniques and the results strongly depend upon the choice of an acceptance condition which is hard to determine and the seed pixel.

### 2.3.4 Segmentation by Edge Detection

In this method, significant intensity changes between two adjacent pixels are interpreted as edges of the objects in the image. It is easy to compute and fast similar to region growing one. The main drawback is lack of connectivity in detected edges or closed regions. Computationally expensive and unreliable post processing methods is needed to connect edges, which are the border of the same object. If choosing inappropriate mask in the smoothing step, the noise may not be reduced or reduced some important information. A few recent advanced segmentation method is discussed in chapter 3.

All the segmentation methods mentioned are belonged to the interactive or semi-automatic segmentation. There are some automated segmeatation methods [59] which apply the syntactic and semantic elements to an image.

A virtual bronchoscopy system needs airway paths for effective use [17] [54] [55]. There are two general approaches for path definition: manual-path definition and recent automated techniques. The former one is time-consuming and error-prone. The user needs to specify key frames, traces path or self-guided navigation. It cannot readily get many paths. The latter one does not use grey-scale information and also error-prone. It includes the methods like segmentation followed by three dimensional skeletonization, active contour models, morphological operations, estimation of principal eigenvectors and vector fields. They require inordinate processing time and some of them lead to imprecise or missing paths. Higgins [55] presents an automatic path-computation method, which is robust to data anisotropy. It uses true grey-scale information and has

very fast computation. Furthermore, it gives many paths. Hong [19] presents an interactive virtual colonoscopy method, which uses a physically based camera control model and a hardware assisted visibility algorithm. A user provides the path starting and ending points, it can automatically determine the path. Users also can interactive navigate in it.

## 2.4 Rendering

This step should be familiar to those people with knowledge of display techniques for volume data. There are two main rendering methods: volume and surface rendering methods [11]. Volume rendering topics includes raycasting volume renderers and splatting. The surface rendering includes marching cubes and the dividing cubes algorithm, polygon decimation and mesh simplification. We should retain our focus on how to best extract pertinent information and present it for maximum effect.

Ramaswamy [44] suggests another coherence-based dynamic navigation method. They present a fast volume rendering method for virtual endoscopic exploration on an inexpensive workstation. For computing a dynamic sequence of views, researchers suggest methods that exploit the temporal coherence between adjacent views. They propose volume-rendering based navigation method exploit the temporal-coherence concept. Brady [5] presents a similar method. It is a two-phase perspective raycasting algorithm that considering the coherence inherent in adjacent frames during navigation.

By using two dimensional or three dimensional textured data slices and a blending operator, three dimensional texture mapping [32] approach is much faster with hardware-accelerated texture than ray casting. This method is a direct data visualization technique that is similar to ray casting. Three dimensional textures are a logical extension of two dimensional textures. In three dimensional textures, texels become unit cubes in texel space. The three di-

mensional texture is used as a voxel cache, processing two dimensional layer each time by all rays simultaneously. It has taken the advantage of spatial coherence. However, the main disadvantage of three dimensional texture compared with CPU based volume technique is the missing capability of shaded rendering. It may be achieved by applying the shaded volume rendering algorithm [9].

## 2.5 Display

Traditionally, the images are viewed on film even for the volume data images. It is caused by the fundamental resolution limitations inherent in CRT displays. Nowadays, there are some medical applications tried to emerge technology in virtual reality. Robb [45] mentions that using of virtual reality, users can be allowed to manipulate with intuitive immediacy similar to that of real objects. The viewer can interactively visualize the illustrated object. Furthermore, he suggests that by engaging other senses such as touch and hearing or even smell to enrich the visualization.

## 2.6 Hazards of Visualization

It is essential that three dimensional medical visualization systems are capable of complex image processing algorithms and provide representations that are more informative than the original. The trained professional should be able to search for cues and structures from the augmented presentation. The followings are the main related hazards [60] that should be noticed when constructing the three dimensional medical data image.

### 2.6.1 Adding Visual Richness and Obscuring Important Detail

Some fancy computer graphics algorithms are easily misleading others. Texture mapping is an powerful technique to increase visual cue, but it may distract and hide certain information about the quality of the tissue of the object. Sometimes the fuzzy pictures and a cloud of glowing voxels convey provide more information than isosurface representations of the same data though without visually appealing images. The most informative visualization is not always the most visually appealing at first glance.

### 2.6.2 Enhancing Details Incorrectly

Data often consists a significant amount of ambient noise. The noise may be removed automatically by the visualization system and a cleaner image is reconstructed. This may cause an error, as the noise may be part of the observable detail. The visualization should always consider the clinical problem first and provide the flexibility for the clinician to decide.

### 2.6.3 The Picture is not the Patient

We often try to improve comprehension by emphasizing a particular feature of an image, and exaggerating the strength of the feature. However, we may mislead the observer into believing what he has observed is the real picture of the patient. We should try to emphasize without betraying the truth.

### 2.6.4 Pictures-'R'-Us

Computer graphics people often tempted to produce pretty pictures. However, we must keep in our mind that we are solving medical problems. We should address the problems of the patients being asked by the clinicians but not to produce entertainment quality images to win any awards.

# Chapter 3

# Overview of Advanced Segmentation Methodologies

To recognize an object in a image, segmentation is the main process. It is an important step for most of analysis procedures. Apart from the basic segmentation methods introduced in section 2.3, some advanced methods are discussed.

## 3.1 Mathematical Morphology

Pisupati [41] [42] proposes an mathematical morphology algorithm to automate the segmentation of pulmonary tree structures in the lung. In mathematical morphology, structuring elements which are geometric structures of different sizes and shapes are used to process images. It is used to investigate geometric structures in images. In the algorithm, it uses different two dimensional morphological operators, i.e. dilation, erosion, opening and closing to analysis the images to obtain potential airways in each image by using the segmented knowledge of previous images to process the current image. As lung tree have almost cylindrical branches, they use circular structuring elements. Grey-scale reconstruction operator is the main operator. Two images are inputed, mask image $I$ and marker image $J$. The previous image is assigned as marker image

$J$ and current image is assigned as mask image $I$. The reconstruction operator extracts connected components of $I$ which are marked by $J$. Then segmented airway volume is obtained by processing three dimensional region growing. The algorithm has an assumption that airway has wall of uniform thickness which may not match with real data. This assumption will result of underestimate or overestimate vascular regions. Using different radius of circle structuring elements will affect the segmented images. In additon, the choices of marker image and mask image are crucial to the result too.

## 3.2 Recursive Region Search

Another method [22] is a kind of recursive search method of cross section of tree structured objects. This method requires a cross section of a trunk of a tree structured object as the initial condition. The method consists of two parts. First it detects the initial cross section of each trunk as the starting point of the proposed method. Second it extracts all branches belonging to it. During the processing, the location of branching points and the numbers of branches at them are memorized. When the top of a branch is reached, the immediate branching point is identified and a search of another branch is repeated. Each cross sectional region is perpendicular to the direction of the branch. It determines the child regions by using a direction vector. It is a vector that is vertical to the parent region and its direction conincides with the direction of the top of the branch. It is a unit vector with magnitude same as the pixel interval of CT image. The regions are thresholded. If proper candidate region is detected, the direction vector is modified to the vector which connects the center of gravity of the parent region and that of the detected child region. The newly child region is assigned as the next parent region and recursive the process. However, in the paper, it does not mention the siturations to determine a proper child region. If two child regions are closed each other, it is a hard problem to determine which is belonged to current parent region as

both have the chances.

## 3.3 Active Region Models

Ivins and Porrill [23] describe a segmentation method for medical images using a closed snake which is driven by a pressure force. Snakes are active contour models and the pressure force is a statistical function for describing the characteristics of image data. Apart from pressure force, there are tension, stiffness and repulsion forces. The first two forces are used to keep the boundary smoothness of the region model. The latter force can prevent the self-intersection. When the snake encounters pixels that are not belonged to user-defined limits relative to a seed region, it stops the expansion and may contract the model if any violation between the limits and the pressure force. Each force has an parameter associated with it to preserve the energy balancing of the changing model. The main different of statistical snake and original snake [24] is that the model of statistical snake can be expanded and contracted considerably during energy minimisation. This method is usually applied on brain MR images. However, to ensure that statistical snake executes successfully, a small part of the snake should overlap the desired feature and suitable parameters are required. Using this approach, it is possible to generate three dimensional images apart from the problems of not routine enough as it is too difficult and time consuming. Also medical images usually have not strong enough edges for the model to reach equilibrium.

# Chapter 4

# Overview of Centerline Methodologies

Centerline has an important role in automatic analysis of various anatomies. It is used in computing edge gradients, searching for border positions, deriving video-densitometric profiles and measurement of vessel diameters. It is the basis for three-dimensional reconstruction of vessel segments or of the entire arterial tree. Nevertheless, centerline is acted as a logical frame for quantitation and navigation. Several methodologies are available for centerline detection.

## 4.1 Thinning Approach

Skeletonization, sometimes referred to as thinning, is a process where objects are reduced to structures of lower dimension while preserving the topology and shape. Thinning method is one of the conventional approaches. For three-dimensional skeletonization, the border points are "peeled off" layer by layer while preserving the topology of the original object. It iteratively converts those foreground border points that are not satisfied certain geometrical and topological constraints to background points.

There are two ways to implement thinning operations, sequential and paral-

lel. Gong and Bertand [15] present a simple parallel three-dimensional thinning algorithm, which conserves medial surfaces. The medial surface is the set of centers of maximal Euclidean balls included in the object. This parallel algorithm is based on some deletion predicates. The method of Ge [12] is a sequential one which define a skeleton as the locus of the centers of maximally inscribed balls (CMBs). They propose to use CMBs as geometrical constraints in the thinning algorithm. First, they use CMBs as anchor points so that the resulting skeleton is medially positioned. Second, they use a faster algorithm for calculating the number of connected components within a neighborhood surrounding each voxel in the entire object. Their method is an improvement of Lee's method [27]. In order to detect CMBs, they first compute the distance transform of a binary volume. The distance transform for a point in the foreground is the distance from this point to the nearest point in the background. There are a few types of distance transform and 3-4-5 Chamfer distance transform is used. The weight is 3, 4 and 5 if the distance between two direct neighbors are 1, $\sqrt{2}$ and $\sqrt{3}$ units apart respectively. By comparing the distance value of each voxel with its neighboring voxels, the CMBs are determined. They divide the neighborhood into eight overlapping octants of 2x2 voxels to compute the number of 26-connected components in a neighborhood. They claim that this calculating the number of connected neighbor algorithm avoids the use of any extra checking necessary in the existing approaches and it does not require any extra data structure. The number of computation is, therefore, kept at a minimum. A path between two points specified is determined by applying the thinning algorithm. Extra loops along this path caused by holes may exist which require user to mark those points on each unwanted branch.

Recently, they propose a modified algorithm [13] to reduce the user interaction. Their algorithm involves three steps. In the first step, they generate a three dimensional skeleton of the binary colon volume using a fast topological thinning algorithm. In the second step, they employ a graph search algorithm to remove extra loops and branches. These loops and branches are caused by

holes in the object which are artifacts produced during image segmentation. To remove extra branches and loops, they convert the skeleton to a weighted graph in which each edge corresponds to a branch, and the weight of each edge represents the size of the corresponding branch. They consider the edges in the graph to be pipes and the weights to be their sizes. The true central path, then, should be the widest pathway between two endpoints since the central lumen is always wider than artificial holes. In graph terms, they define the flow of a path between two vertices as the smallest weight along the path. Then the central path is the path with the largest flow. After extra branches are pruned, the skeleton becomes a single path lying medially in the colon lumen. However, the path is usually jagged due to the discrete nature of the CT image data. In the final step, they compute a smooth representation of the central path by approximating the skeleton with cubic B-splines. This final step is necessary because the skeleton contains many sudden changes in direction due to the discrete nature of image data. The user is required to supply two endpoints for the central path. The algorithm is not fully automatic. On an SGI system with an R10000 CPU without optimization, the time ranges from less than 1 minute for a subsampled 10MB volume to several minutes for a 150MB volume using the original resolution. The time required to convert a skeleton to a graph and to search for the true central path never exceeded 15 seconds. The time required for relatively small volumes (100MB) is acceptable for practical applications. However, speed remains an issue since a large volume (200MB) may take about 10 minutes of total time to process.

Nystrom [38] describes another skeletonization applied to magnetic resonance angiography images. The first skeletonization step reduces the object to a surface skeleton from which the original object can be recovered. The surface skeletons are computed in two phases. During the first phase, removable voxels are identified and iteratively removed until an at most two voxel thick surface of skeletal voxels is identified. During the second phase, this set is reduced to unit-wide surfaces and curves in a thinning process that has to be split into

six (the number of face-neighbours of a voxel) directional processes, each of them applied once. The second step reduces the surface skeleton further to a curve skeleton in two iterative phases. Voxels are iteratively removed from the surfaces and reduced to unit thickness by iteratively applying a final thinning process, until no more voxels can be removed. To obtain cleaner skeletons an optional pruning step can follow the skeletonization. The skeletons are labelled with the distance to the original background. Using this information, a simple and brute-force pruning method is applied. The shortest skeletal branches, labelled with small distances, are probably resulted from either noise or weak protrusions of the object. They can be pruned in iterated scans. Every skeletal voxel with a distance label less than a threshold is removed, unless its removal would disconnect the skeleton. For a 30 Mbyte MR angiography image the computation times for the whole process take about 15 minutes.

## 4.2 Volume Growing Approach

In Samara's paper [48], the centerline calculation technique is based on a volume growing method that segments a volume corresponding to air in CT data obtained from an insufflated colon. A 10-point neighborhood consisting of eight in-slice voxels (adjacent in-slice voxels including the diagonal voxels) and two out-of-slice voxels (directly above and below the voxel of interest) is used for volume growing. Volume growing is initiated by indicating a seed point in the rectum, air voxels are grown and tagged with growth step numbers. Each grown voxel is assigned an iteration number one more than its parent voxel corresponding to the step of growth from the seed voxel. The seed voxel is initialized to a growth step number of zero. Voxels with the same growth step number lie along a wave front. The centers of mass of grown voxels with similar growth step numbers are used as a 'forward centerline'. The last voxel added to the volume grown during this initial volume growing is used as a seed voxel for a 'backward centerline' volume growing. This procedure is repeated by growing

from the cecum to the rectum to generate a 'backward centerline'. Each point on the forward centerline is averaged with its nearest neighbor on the backward centerline to yield a point on the final calculated centerline. The technique is evaluated on a clinical colon case by comparing the calculated centerline with points indicated by two radiologists. Root mean square (RMS) differences between the computed and indicated points are small (4-5mm) and comparable to inter-observer differences. Results indicate that with this technique the centerline of the colon can be accurately and quickly calculated. The time required for centerline computation is approximately one minute on a Silicon Graphics O2 workstation. The average RMS difference between the indicated and calculated points is 4.924 mm.

## 4.3 Combination of Mathematical Morphology and Region Growing Schemes

Masutani [31] proposes a new model based on region growing controlled by math-morphological information of local shape, which has the ability of topological correction to extract bifurcation structure of three dimensional vascular. First, the initial shape is acquired by thresholding. Then, region growing is processed basically in the space limited by the initial shape. Structural information of the shape is simultaneously obtained. The region grows while avoiding non-vessel regions and keeps its local smoothness based on math-morphological information and local shape processing. Mathematical morphology is a well-known theory for structure analysis and processing of binary shapes. Most math-morphological operations and analysis methods can be defined in terms of the basic operations erosion, dilation, closing and opening. One of the most important characteristics of them is controllability of their results by size and shape of filter kernel (structuring element). Morphological size is defined as a math-morphological quantity, which is an attribute of voxels that belong to an object. Distribution of morphological size in a shape is called pattern spec-

trum to represent the characteristics of the whole shape. Morphological size of a voxel in an object means the minimum size of kernels, which can remove the voxel from the shape by opening operation. It represents local thickness of the object at the position of a voxel. After the process, segments of small region (cluster) are produced. Each cluster's center of gravity [30] will be used as the approximation for a node position. Centerline can be reconstructed by linking the nodes created. In this algorithm, except for initial shape acquisition and seed definition, users require to select a suitable mode. The growth termination condition is also variable.

## 4.4 Simultaneous Borders Identification Approach

In Sonka's paper [52], it mentions a semi-automatic lumen centerline detection in complex coronary angiograms. The method is based on simultaneous detection of the approximate position of the left and right coronary borders. There are four stages: observer identification of segment of interest, straightened edge image generation, simultaneous detection of approximate coronary and lumen centerline detection. In the first stage, the observer needs to define at least two points located within the vessel lumen which indicate the region of interest, (which segment of coronary). The two points serve as starting point and ending point. Two additional points are entered, one on either side of the vessel, to define the width of the image region to be analyzed. A estimated starting centerline is produced and then smoothed. In next stage, the vessel segment of interest is straightened and smoothed in a direction parallel to the centerline estimated. In the third stage, the simultaneous border detection method [51] uses a three dimensional graph searching approach. The optimization criterion used incorporates information about the global characteristics of the border pair into the detection process. The cost function used causes a high contrast border to more strongly influence the position of the opposite border than a low contrast border does. It identifies the border pair that has the greatest proba-

bility of corresponding to the actual coronary borders. Finally, the midpoints between the left and right borders are calculated in the straightened coronary and are mapped and then smoothed in the original image space. Comparing with other conventional detection, the computational complexity is significant increased. In addition, it requires manual definition of the vessel centerline together with two additional points that serve with the centerline to define the vessel segment to be analyzed.

## 4.5 Tracking Approach

In the paper of Swift [55], they provide an axis generation method for three dimensional virtual bronchoscopic image. It deals with three dimensional branching tubular structures to find smooth central axes through the major airways. The user specifies a starting point based on the interested structure (i.e. the trachea). Then using an adaptive searching technique, their method automatically steps through the structure and recording its movement and making measurements along the way. For each point during the searching, the airway cross-section is determined through a combination of grey-scale image data and a prior structural model constraints. After getting the first seek point as a viewing point, it casts rays radially outward to estimate the radius of a circle enclosing the tubular structure and the locations of tube walls. The centroid of the structure's cross-section is calculated from estimated wall locations. New candidate branches are determined and the new corresponding viewing points are produced. Based on the cross-section estimated of every viewing point, a path is determined. It requires a few minutes for a typical 512x512x25 CT image.

## 4.6 Distance Transform Approach

Niblack [37] uses distance transform to generate skeletons and centerlines. The main advantage of this approach is that they are non-iterative. The skeleton is produced in a fixed number of passes through the image regardless of the object sizes. They first compute the distance transform, then select a subset $M$ of pixels which is the set of local maxima in the distance transform and finally select pixels to connect the pixels, called saddle points, whose neighborhoods contain alternating humps and valleys. An example of saddle point is a pixel with distance transform value $x$ such that clockwise around its four neighbor pixels there are pixels with values $> x$ (part of a hump), $< x$ (valley), $> x$ (hump) and $< x$ (valley). The set of pixels in $M$ is connected by climbing "uphill", i.e. along paths of increasing distance transform values, from saddle points and neighbors of local maxima. For a 528x512 test image of a printed circuit board from an industrial application which contain several thick (approximately 90 pixel diameter) circuit pads, using a contour-based distance transform, their algorithm requires 2.5s. The experiments are run on an IBM RS/6000 with approximately 20 mips.

Blezek [2] uses a distance transform (three dimensional chamfer 3-4-5 distance map) to label each voxel in the object with a medialness measure. The shortest path between start and end positions is found using Dijkstra's single-source shortest path algorithm using the medialness measure as weights in a graph search algorithm. The 3-4-5 chamfer distance transform closely approximates the true Euclidean distance transform and is computationally efficient. Each voxel is considered as a node in a weighted, directed graph and have 26 neighbors which are connected by edges. The weight or distance associated with moving from any neighbor voxel to an adjacent voxel is a function of the adjacent voxel's medialness measure. The path of "least resistance" through voxels that are more medial is determined. Our algoirthm is based on Blezek method which he applies to virtual endoscopy system. Endoscoopy system is

for examining visually the interior of a bodily canal or a hollow organ such as the colon, bladder, or stomach. Bronchoscopy system is for inspection of the interior of the bronchi. Bronchoscopy system and endoscopy system differ in the applied object. The latter one usually is applied on hollow organs and cavities without special consideration of the branching feature. In virtual bronchoscopy system, careful consideration of the branching characteristic is required. The comparison between our algorithm and his algorithm is given in section 7.2 in chapter of Experiments and Discussion.

# Chapter 5

# Automated Extraction of Bronchus Area

## 5.1 Basic Idea

The bronchus is bounded by a thin wall and is filled with air. The CT values (intensity) of the wall and the bronchus area have very high contrast since air has low CT values and the wall has much higher CT values comparatively. Instead of extracting the wall of the bronchus as curved surfaces, the inside area of the bronchus is extracted as three dimensional thin volumetric areas. The extracted areas should be in tree shape. Based on the simple fact that the bronchus area is simply connected, the extraction procedure is performed by tracing voxels with relatively low CT values corresponding to air. The process is controlled so as not to proceed across voxels with relatively high CT values corresponding to the wall. The results of region growing strongly depend upon the choice of the acceptance condition and the initial point. Therefore, a suitable start point of tracing and the criterion to merge a new area should be selected carefully.

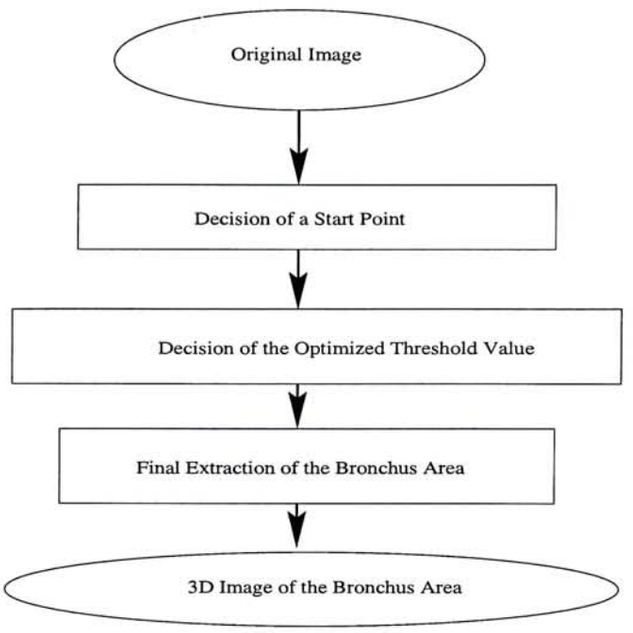



**Figure 5.1**: The Outline of the Bronchus Area Extraction Algorithm

## 5.2 Outline of the Automated Extraction Algorithm

Given a CT image, the automated bronchus extraction algorithm produces three dimensional image of the bronchus area. Processes of the algorithm include decision of a start point, decision of the optimized threshold value, and final extraction of the bronchus area. The outline of the bronchus area extraction is shown in Fig. 5.1.

### 5.2.1 Selection of a Start Point

In a typical image of lung, we have different regions of interest (ROI) of almost circular/elliptical, which correspond to the cross-sections of the tree branches [41]. We want to retrieve an inside point of bronchus that is a point among the cross-sections of the bronchus area as an initial point for region growing. Moreover, an inside point of trachea, which is located near the center of the image, is chosen in order to obtain a better result. In order to meet all these criteria, a start point retrieval algorithm based on genetic algorithm (GA) is proposed. The detailed description of the algorithm will be presented in section

5.3.

### 5.2.2 Three Dimensional Region Growing Method

Given a start point, the region growing method checks its neighbors to determine whether they are belonging to the same region. If they are determined to be one of its connected area, their neighbors will be checked. The process is recursively executed until no more new neighbor is merged. For a three dimensional space, the six neighbors of a point are to be checked. The three dimensional painting algorithm [35] is applied with some modifications for optimization purpose and will be described in section 5.4.

### 5.2.3 Optimization of the Threshold Value

Before applying the above three dimensional painting algorithm, the threshold value for distinguishing the bronchus wall area from the inside of the bronchus area must be carefully chosen. If the threshold value is too low, the extracted bronchus area will be underestimated. On the other side, if it is too high, the extracted bronchus area will not be accurate enough to represent the actual bronchus. However, based on the feature of the extracted area number of different threshold values, we can determine the optimal threshold value. It will be explained in the section 5.5.

## 5.3 Retrieval of Start Point Algorithm Using Genetic Algorithm

In the extraction of bronchus from three dimensional CT images of lung, the most common and effective method is three dimensional region growing. Given a start point, the algorithm traces voxels with relatively low CT values corresponding to air without proceeding across voxels with relatively high CT values

corresponding to the wall. It is obvious that a suitable start point should be selected carefully.

The start point determination method, which is based on a cost-minimization approach, is implemented by using genetic algorithm (GA) [47] to accurately locate a start point for the region growing. A start point should be an inside point of trachea and near the center of the human body. In addition, it should have reasonable connected region number. To simplify the problem, we transform the problem from three dimensional to two dimensional. Furthermore, we assume that the images are axial and the first slice can cut the main bronchus. We determine the start point by considering the first slice only.

### 5.3.1 Introduction to Genetic Algorithm

Genetic Algorithm [47] can be applied to any problem that has a large search space. It searches directly in the space of individuals, with the goal of finding one that maximizes/minimizes the fitness function. The search can be in parallel because each individual in the population can be seen as a separate search. It is hill-climbing because we are making small genetic changes to the individuals and using the best resulting offspring. We should concentrate on the most promising individuals, but if we ignore the low-scoring ones, we risk getting stuck on a local maximum. The fitness function depends on the problem, but in any case, it is a function that takes an individual as input and returns a real number as output. In the "classic" genetic algorithm approach, an individual is represented as a string over a finite alphabet. Each element of the string is called a gene. The selection strategy is usually randomized, with the probability of selection proportional to fitness. Usually, selection is done with replacement, so that a very fit individual will get to reproduce several times. Reproduction is accomplished by cross-over and mutation. First, all the individuals that have been selected for reproduction are randomly paired. Then for each pair, a cross-over point is randomly chosen. Each gene can be

altered by random mutation to a different value, with small independent probability. Like neural networks, genetic algorithms are easy to apply to a wide range of problems. Furthermore, genetic algorithms can be incredibly efficient if programmed correctly.

### 5.3.2 Problem Modeling

We want to extract the three dimensional structure of bronchus in lung area by the region filling method. In this method, it is important to obtain a good start point so that we can extract the necessary information effectively and efficiently.

Simply speaking, coordinates of the start point serve as the inputs of our cost function, and the problem here is to find the point which has the minimized cost. A point consists of $x$-coordinate and $y$-coordinate data which form a tuple $(x, y)$. The possible value of $x$ is within the range of zero to the width of the given image. The possible value of $y$ is within the range of zero to the height of the given image. Therefore, we have a (width x height) solution space. For example, the sample space of a 256 x 256 image has 65536 combinations. Our problem that has a large search domain is suitable to be tackled by genetic algorithm.

For the purpose of constructing our model, we introduce the chromosome encoding and the fitness function. A chromosome is represented by two genes. The first gene is represented by the $x$-coordinate while the second gene is represented by the $y$-coordinate. Therefore, two genes are represented as a point. There are two cost factors: the connected region cost and the near center cost. Each cost has its corresponding cost weight which controls the relationship with the costs.

As mentioned above, a proper start point should have reasonable connected region. It should have less than 0.5% of the total area. For example, in a 256 x 256 image, it should not have connected area which is greater than 300.

The first cost is the difference between the reasonable connected area and the connected area determined by applying a four-direction flood filling algorithm from the given point. Therefore, the first cost is $|reasonable\_connected\_area - connected\_area\_determined|$.

The second cost is the distance from the current point to the center point of the image. The center point of the image is calculated as $(dimension_x/2, dimension_y/2)$. Therefore, the distance is $\sqrt{(x - center_x)^2 + (y - center_y)^2}$.

The corresponding weights are determined by the fact that the ratio of first cost to the second cost is 100:1. Since it is a cost minimization problem, the two costs are divided by the corresponding maximum possible values. The maximum value of the first cost is the total area, which is the dimension of $x$-direction multiplies the dimension of $y$-direction. The maximum value of the second cost is the distance from any corner to the center point. Therefore, the first weight is $100/(dimension_x \times dimension_y)$, and the second weight is $1/\sqrt{(dimension_x/2)^2 + (dimension_y/2)^2}$. For a 256 x 256 image, the two weights are 100/65536 and 1/181 respectively.

The fitness function of a given population $(x, y)$ is

$$\mathcal{F}(x, y) = \sum_{i=1}^{2} \omega_i \mu_i$$

The connected cost is

$$\mu_1 = |reasonable\_connected\_area - connected\_area\_determined|$$

The weight constant of connected cost is

$$\omega_1 = \frac{100}{(dimension_x \times dimension_y)}$$

The near center cost is

$$\mu_2 = \sqrt{(x - center_x)^2 + (y - center_y)^2}$$

The weight constant of near center cost is

$$\omega_2 = \frac{1}{\sqrt{(\frac{dimension_x}{2})^2 + (\frac{dimension_y}{2})^2}}$$

### 5.3.3 Algorithm for Determining a Start Point

1. Generate initial populations ($n$=100)
2. do {
3. Perform mutation with 0.05 probability
4. Perform crossover with 0.05 probability
5. Evaluate the fitness of each population
6. Rank the populations
7. Keep the best $n$ populations
8. } while (termination is $FALSE$)
9. Output the best one

### 5.3.4 Genetic Operators

The population size is 100. Since the potential candidates are located near the center of the image, the initial population is selected uniformly from the center region of the image.

A simple mutation operation can free a chromosome from a local maximum. The mutation operation is performed by picking up a chromosome randomly. We can create the four additional chromosomes by the following method: adding the first gene by one, adding the second gene by one, subtracting the first gene by one and subtracting the second gene by one. The mutation probability is 0.05. Given population size is 100, there are 5 mutation operations and each produces four children. Therefore, 20 children are produced.

The crossover operation can generate more diversified offspring. To perform the crossover operation, two chromosomes are picked up randomly and then their genes are exchanged. The crossover point is located between the first gene and the second gene. The probability of crossover is 0.05. For a population size of 100, 5 crossover operations are performed and each produces two children. In each round 10 children are added to the population.

After the mutation and crossover operations are completed, all the chromosomes in the population that includes the 30 children, which have just been added to the population, are ranked by the fitness function. Only the first best $n$ chromosomes in the population are kept for next generation. The algorithm terminates when the generation number exceeds the maximum generation number.

## 5.4 Three Dimensional Painting Algorithm

Three dimensional painting algorithm [35] is a three dimensional region growing method [59] and it paints all the voxels that have lower CT values of the given threshold value to the same color starting from the given initial point. The painted voxels are corresponded to the connected bronchus area. The algorithm is performed sequentially, line-by-line mode along axes in three dimensional space to reduce the computation time. Starting at the given initial point given that must be an inside point of the bronchus area, it searches all its adjacent points to determine whether they have the CT values lower than the given threshold value. If the visited voxel has a CT value lower than the given threshold value, it is recognized as an inside point of bronchus.

### 5.4.1 Outline of the Three Dimensional Painting Algorithm

1. Load the original image on $F = f_{ijk}$
2. Insert the start point to $Q$ (a FIFO type Queue)

3. While ($Q$ is not empty) {
4. Get the first point from $Q$ and store it to $(x, y, z)$
5. while ($f_{xyz} < threshold$)
6. $x = x - 1$;
7. $p_x = x; p_y = y; p_z = z$;
8. $length$=0;
9. while ($f_{xyz} < threshold$) { // count the length to paint
10. paint $g_{xyz}$ //the output image, initial all are non-painted
11. $length$++; $x$++;
}
12. $x = p_x; y = p_y - 1; z = p_z$;
13. if ( ($f_{xyz} < threshold$) and ($g_{xyz}$ not paint) and (($x, y, z$) not in $Q$) )
14. insert $(x, y, z)$ to $Q$
15. $x$++;
16. for ($i$=0 to $length$) {
17. if ( ($f_{x+1,y,z} \geq threshold$) and ($f_{xyz} < threshold$)
18. and ($g_{xyz}$ not paint) and (($x, y, z$) not in Q) )
19. insert $(x, y, z)$ to $Q$
20. $x$++;
}
21. $x = p_x; y = p_y + 1; z = p_z$;
22. repeat step 10-12
23. $x = p_x; y = p_y; z = p_z - 1$;
24. repeat step 10-12
25. $x = p_x; y = p_y; z = p_z + 1$;
26. repeat step 10-12
}

Given an input image, initial point and threshold value, the algorithm starts with putting the initial point in a First-In-First-Out (FIFO) type queue. While the queue is not empty, the following operations are performed. Firstly, it gets

a point from the queue as a reference point and scans left to the first point which is needed to be painted along the $x$-axis. Starting from the leftmost point, it paints all the following points that have CT value lower than the threshold value along the $x$-axis and the number of painted points is recorded. Then, it determines all the changing points by scanning the points above the reference point in the $y$-direction with the recorded painted length and add them to the queue. A changing point is a point that has the CT value lower than the threshold value and one of its next point along the $x$-axis has the CT value higher than the threshold value. All the changing points that above and below the reference point in the $y$-direction and $z$-direction are obtained and added again. To optimize the algorithm, any point that is already in the queue will not be added to the queue. Every point should be distinct in the queue at a time, i.e., no point should be repeated in the queue. The algorithm terminates when the queue is empty.

## 5.5 Optimization of the Threshold Value

An optimal threshold value is the CT value that can distinguish the bronchus wall area from the inside of the bronchus area. It can decide whether a current voxel should be merged into the colored area or not. It is desirable that the threshold value is as high as possible for extracting region objects with minimum number of misrecognition. However, if the threshold value is higher than the optimal one, the extracted region may grow over the bronchus wall and the whole lung area will be extracted as the bronchus area in the worst case. This phenomenon is called 'explosion.' Since the lung volume is much larger than the bronchus volume, the number of extracted voxels of the explosion case is much larger than that of the normally extracted bronchus area.

We start the painting with a relatively low threshold value. Number of voxels extracted as the bronchus area during the operation is counted. The painting is iterated with another higher threshold value by increasing the pre-

vious threshold value with an appropriate increment. After we complete the experiments, a threshold value that just before the point on which the number of voxels extracted is increased explosively is regarded as the optimal threshold value, i.e., it is just before the point which has the maximum rate of change of the slope of the graph plotting number of voxels extracted against threshold value. We can obtain the result by calculating the difference between two adjacent second derivatives of the curve and the point which has the maximum value regarded as the optimal threshold value.

# Chapter 6

# Automatic Centerline Determination Algorithm

Our automatic centerline determination algorithm is composed of two main components. Fig. 6.1 shows the work flow of our algorithm. 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 reads 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.

## 6.1 Distance Transformations

A distance transformation [3] [4] is an operation that converts a binary image to a grey-level image where all pixels have a value corresponding to the distance to the nearest feature pixel. Computing the distance from a pixel to a set of feature pixels is essentially a global operation which are prohibitively costly. Therefore, algorithms should consider only small neighborhood at a time and still maintain a reasonable approximation to the Euclidean distance. Distance transform, (DT), based on the idea of global distances in the image which are approximated

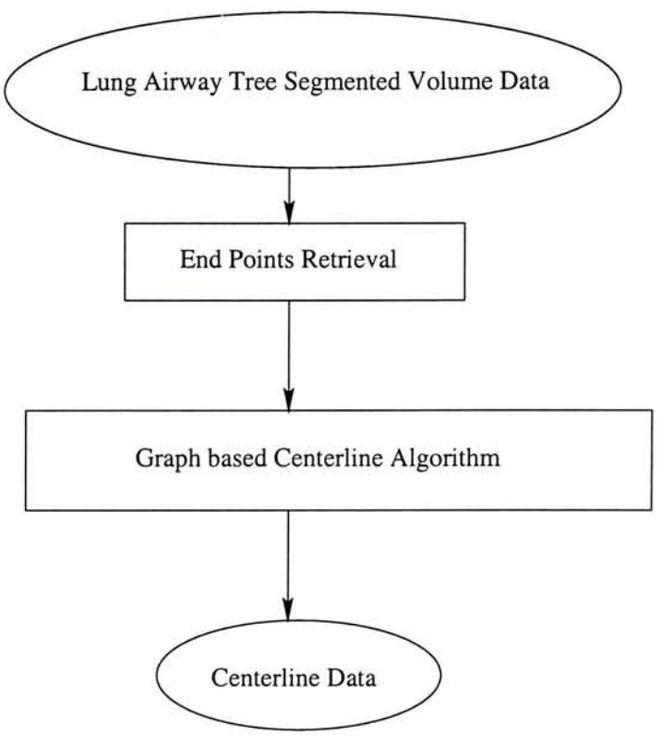



**Figure 6.1**: Outline of the Automatic Centerline Determination Algorithm

by propagating local distances, i.e., distances between neighboring pixels. This propagation can be done either in parallel or sequentially. An original binary image consists of feature pixels with the initial value zero, and non-feature pixels with the initial value infinity, i.e., a suitably large number. All DTs can be described in graphical form as "masks". The DT masks are not linear filters. The constants are the local distances that are propagated over the image. The size of the neighborhood can vary.

The computation of the DT is either parallel or sequential. In the parallel case the center of the mask is placed over each pixel in the image. The local distance in each mask-pixel is added to the value of the corresponding image pixel. The new value of the image pixel is the minimum of all the sums. The process is repeated until no pixel value changes, i.e., the number of iterations is proportional to the largest distance in the image. The sequential algorithm also starts from the zero/infinity image. The symmetrical parallel mask is split into two masks. The masks are passed over the image once each: the forward one from left to right, and from top to bottom, and the backward one from right

to left and from bottom to top. The new value of the "central" image pixel is minimum of the sums of the image transform is computed. The final DT result is exactly the same whether the parallel or sequential method is used.

Three-dimensional elements are usually called voxels (volume elements). Each voxel has 26 neighbors which can be classified into three categories. The first, closest, kind of neighbors is the six ones joined to the voxel by a plane, the second kind is the 12 neighbors joined by a line, and the third kind is the eight neighbors joined by only a point. The three different local distances to the different kinds of neighbors are denoted by $d_1, d_2$, and $d_3$. The algorithms can be illustrated by masks in three dimensions which marks are now parts of a $3^3$ cube in Fig. 6.2. The two planes of each mask are placed close together in the figure. Two passes over the volume are needed. The forward mask is moved over the volume from left to right, top to bottom, and front to back. The backward mask is moved in the opposite way. In each position, the sum of the local distance in each mask-voxel and the value of the voxel it covers is computed, and the new value of the zero voxel is the minimum of these sums.

The three dimensional sequential DT algorithm is:

$distance\_transformation\_3D()\{$
Forward:
for $k = 1$ to $z\_res$
for $j = 1$ to $y\_res$
for $i = 1$ to $x\_res$
$v_{i,j,k} = minimum(v_{i+l,j+m,k+n} + d_{forward\_mask}(l, m, n))$,
$(l, m, n) \in$ forward mark
Backward:
for $k = z\_res$ to 1
for $j = y\_res$ to 1
for $i = x\_res$ to 1
$v_{i,j,k} = minimum(v_{i+l,j+m,k+n} + d_{backward\_mask}(l, m, n))$,

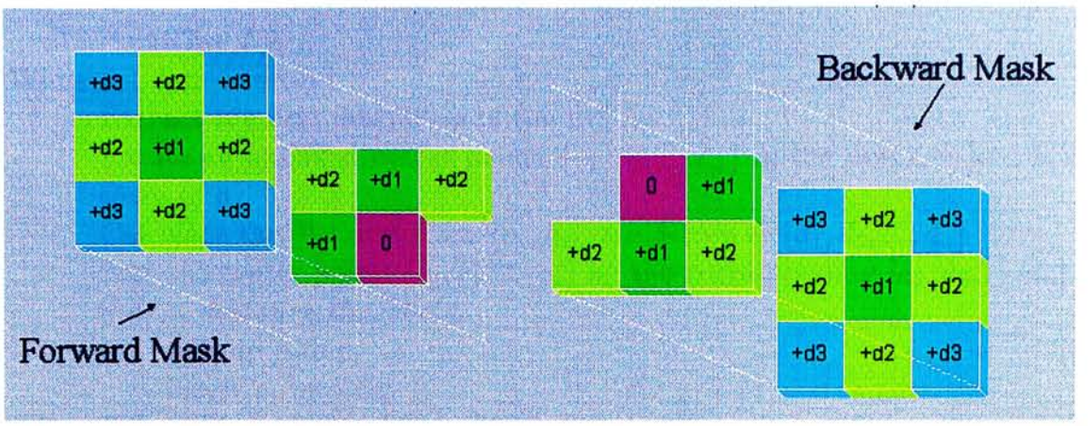



**Figure 6.2**: Masks for the three dimensional distance transform algorithm

$(l, m, n) \in$ backward mark

}

where *x_res*, *y_res* and *z_res* represent the resolutions of the image in $x$, $y$ and $z$ directions; $v_{i,j,k}$ is the value of the voxel in position $(i, j, k)$ in the object volume; $(l, m, n)$ is the position in the mask, i.e. (0,0,0) represents the center of the mask and $d_{forward\_mask}(l, m, n)$ and $d_{backward\_mask}(l, m, n)$ are the constants from the mask, i.e. $d_1$, $d_2$ and $d_3$. The values of $d_1$, $d_2$ and $d_3$ are set to one to infinity depending what kind of DT. If it is set to infinity, it means that the corresponding mask-voxel is omitted from the mask. It is often desirable to use only integers. A very good integer approximation of the optimal local distances is $d_1 = 3$, $d_2 = 4$, and $d_3 = 5$. This chamfer 3-4-5 distance is used in our algorithm.

## 6.2 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 categories, feature or non-feature pixel. The feature pixels

```
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 {c1, c2, ... cn} in the previous and next
        slices of p
4.      Calculate the distance from {c1, c2, ... cn} to p as [d1, d2, ... dn]
5.      Sort the center points [c1, c2, ... cn] by the corresponding [d1, d2, ... dn]
6.      if d1 ≤ distance_threshold
7.          link up p to c1
8.          insert c1 into list L
9.      if d2 ≤ distance_threshold
10.         link up p to c2
11.         insert c2 into list L
12.     if d1 or d2 > distance_threshold and ≤ 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 (c1 or c2)
        }
    }
```

**Figure 6.3**: Center Points Linking Algorithm

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 which is shown in Fig. 6.3. Based on the bifurcation characteristics of lung tree volume, we link up those center points starting from the main bronchus to tiny airway tips. By the algorithm in section 5.3 [26] 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 $\{c_1, c_2, \ldots c_n\}$ be the set of unlinked center points which are in the previous or next slices of $p$. Distance between $p$ and all elements in $\{c_1, c_2, \ldots c_n\}$ will be calculated so that any point will be linked up to $p$ if it is close to $p$.

Let $x_l$ and $x_{l+1}$ be two center points in $l^{th}$ and $l + 1^{th}$ slices respectively. There are two threshold values *distance_threshold* and *possible_distance_threshold* used in our algorithm which measure the closeness between two points. If the distance between $x_l$ and $x_{l+1}$ is not larger than *distance_threshold* and $x_l$ has less than two 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 child-parent 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 only have one parent. Once it is selected from a point, the child-parent relationship is established and no other point can be its parent afterwards.

The center points linking algorithm is shown in Fig. 6.3. 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.

## 6.3 Graph Based Centerline Algorithm

Based on chamfer distance transform and Dijkstra's single source shortest path algorithm [57] [7], Blezek [2] applies a centerline algorithm to virtual endoscopy. However, he mainly concerns one-start-point-one-end-point endoscopy such as colonary and user is required to supply the start and end points. Virtual bronchoscopy which has many branches cannot directly apply his algorithm. Automatic algorithm is preferred. Therefore, modifications are accomplished.

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.

$$V(x, y, z) = (V_{max} - V_{original}(x, y, z))^2$$

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 that more interior voxels have smaller distance values and more near 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 (vertices). Edges are created between a vertex and its 26-connected neighbours. The weight of a edge is assigned as the distance value of the voxel in in-direction. For example in Fig. 6.4, the weight of edge $(V_0, V_1)$ is four which is the distance value calculated of the corresponded voxel of $V_1$.

A general method solving the single-source shortest-path problem is known

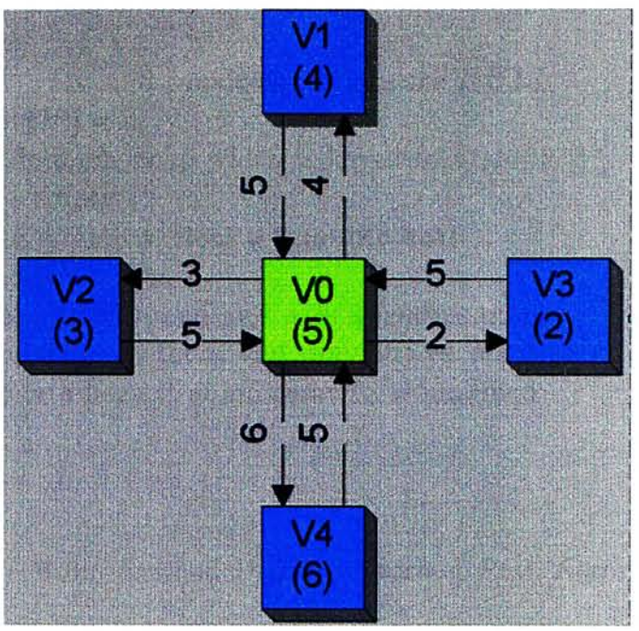



**Figure 6.4**: A two dimensional graph example. The values in the brackets are the corresponding distance value of the voxels.

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 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 [57] [7]. 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 in position $2i$, the right child is in the cell after the left child $(2i + 1)$, and the parent is in position $i/2$. Thus not only links are not required, but the operations

```
dijkstra(G, StartPt){
1.  initial distance[i] = ∞ 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
            }
    }
```

**Figure 6.5**: Modified Dijkstra Shortest Path Algorithm

required to traverse the tree are extremely simple and likely to 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. 6.5 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.

In principle, the required centerline can be constructed by extracting the shortest path between the source point and the end points. It can be accomplished by our algorithm which traces each end points back to source point. In addition, all visited voxels will be marked and will never be visited again so as to speed up the whole tracing process. Combining all visited voxels forms the

resultant centerline of the object. The algorithm for tracing the centerline is given as follows:

```
trace_centerline(){
    initialize centerline(i) as FALSE
    for each current_endpoint ∈ endpoints_list{
        current = current_endpoint
        while current is not NULL{
            if centerline(current) == TRUE
              break
            else
              centerline(current) = TRUE
              current = current → previous_pt
            }
        }
    }
```

In the algorithm, $centerline(i)$ is used to record the tracing state of the corresponding voxel $i$. During the tracing process, if the state is set to be $TRUE$, it means that the voxel $i$ is traced and is marked as a part of the overall centerline. Once the voxel which has $TRUE$ state is set, it will not be traced again and it will not change state afterwards. If it is in $FALSE$ state, it does not necessarily mean that it does not belong to the overall centerline. It only means that it is not included as a part of the centerline up to the current stage. If $centerline(i)$ remains in $FALSE$ state eventually, the voxel $i$ is regarded as a non-centerline voxel. After the process terminates, the resultant centerline can be reconstructed by using those $TRUE$-state voxels.

# Chapter 7

# Experiments and Discussion

Various sets of experiments are performed so as to prove the accuracy and efficiency of our proposed algorithms. All experiments are performed on Sun UltraSparc 5/270 workstation with Solaris 2.6 OS, 512MB RAM and 100Mbps network speed.

## 7.1 Experiment of Automated Determination of Bronchus Algorithm

The automated extraction of bronchus algorithm is executed with real three dimensional CT image data. The image set contains 184 images where each of them is a 256x256 8-bit grey-scale image. In addition, a reduced resolution image set which is composed of 64 128x128 8-bit grey-scale images is prepared for comparison purpose.

In the start point retrieval algorithm, the threshold value is set to be 70. Given the population size is 100, the probability for performing mutation and crossover is 0.05. About 20 generations, the population starts to converge. After another 40 generations, the population converges to a small region. The algorithm terminates when the generation number reaches the maximum population number which is 100. The resultant point, which corresponds to the

best chromosome in the population, is found to be within the required bronchus area.

Different threshold values are applied to the three dimensional region growing algorithm. We test all possible threshold values starting from the lowest value to the highest value (i.e., 0 to 255). Fig. 7.1 and Fig. 7.2 are the corresponding extraction volume results after they are applied to the two data sets. In Fig. 7.1, there are two increments in the graph. The first one which has more dramatic change at threshold 71 is caused by the great volume difference between the bronchus volume and the lung volume. The second increment which is located at the threshold value 150 is due to the excessive high intensity background volume surrounding the lung. However, in Fig. 7.2, the graph of the small data set data only gives one dramatic increment at threshold value 66. It does not have any second increment which is caused by the background region. It can be explained as the excess background region of the small data set is already cropped away during the reducing process. The optimal thresholds are found to be 71 and 66 for the large and the small data sets respectively at which they give dramatic increments. Fig. 7.3 and Fig. 7.4 are the corresponding computation time of the above cases.

It is observed that the two time graphs are decreased gradually starting from around threshold value 150. Before that values, the curve matches the extracted voxel number graphs in Fig. 7.1 and Fig. 7.2. The more voxel number is extracted, the more time is required. However, for the threshold value larger than 150, the number of extracted voxels are mostly related to the background volume and consequently less changing points are produced during the three dimensional region growing. A changing point in the three dimensional painting algorithm is a point that has the CT value lower than the threshold value and one of its neighbor along the x-axis has the CT value higher than the threshold value. The time required for the three dimensional painting algorithm is related to the total number of changing points. Larger number of changing points means that more execution time is required in order to complete the process

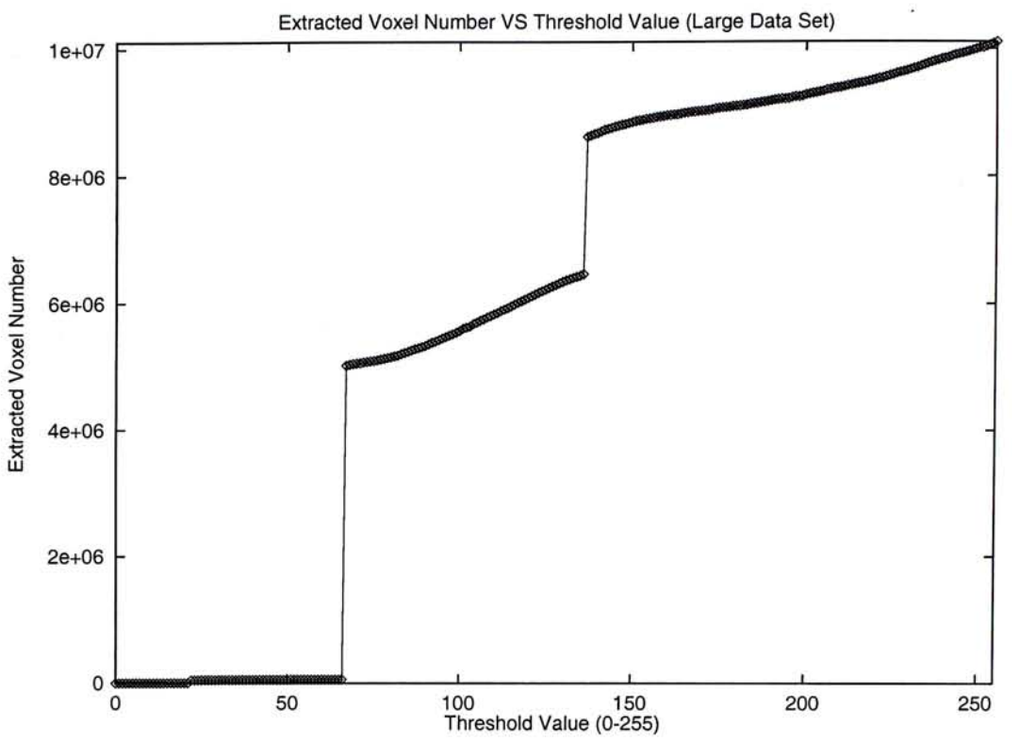



**Figure 7.1**: Extracted Voxel Number in Large Data Set (256x256x184)

as more points are manipulated for producing the region growing. When the threshold value is near to the background intensity, it includes most voxels in the data set and only little amount of remaining voxels have larger threshold values than it. As a result, there are lesser changing points and faster execution time when the threshold value is closer to the maximum threshold value (i.e. 255).

The extracted bronchus areas are visualized with three dimensional texture mapping technique [32], which is a direct data visualization technique that is similar to ray casting. Three dimensional textures are a logical extension of two dimensional textures. In three dimensional textures, texels become unit cubes in texel space. The three dimensional texture is used as a voxel cache, and it processes two dimensional layer each time by all rays simultaneously. Fig. 7.5 and Fig. 7.6 show the images of the reconstructed bronchus. The bronchus are extracted satisfactorily.

Although the two data sets used in experiments represent the same lung

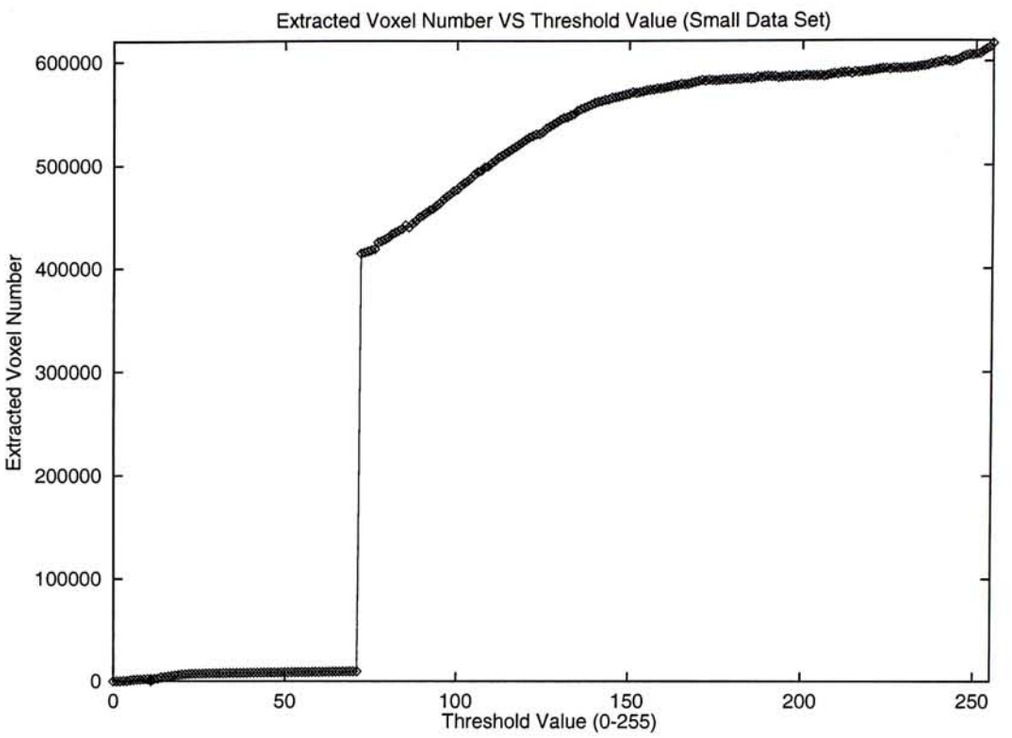



**Figure 7.2**: Extracted Voxel Number in Small Data Set (128x128x64)

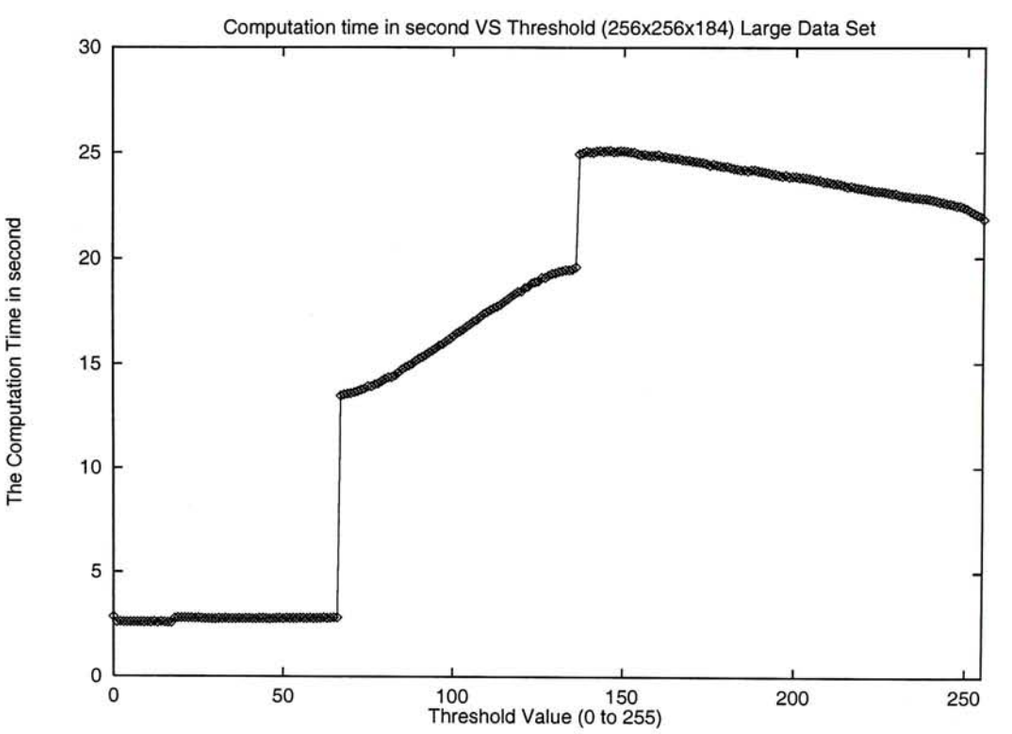



**Figure 7.3**: The Computation Time in Large Data Set (256x256x184)

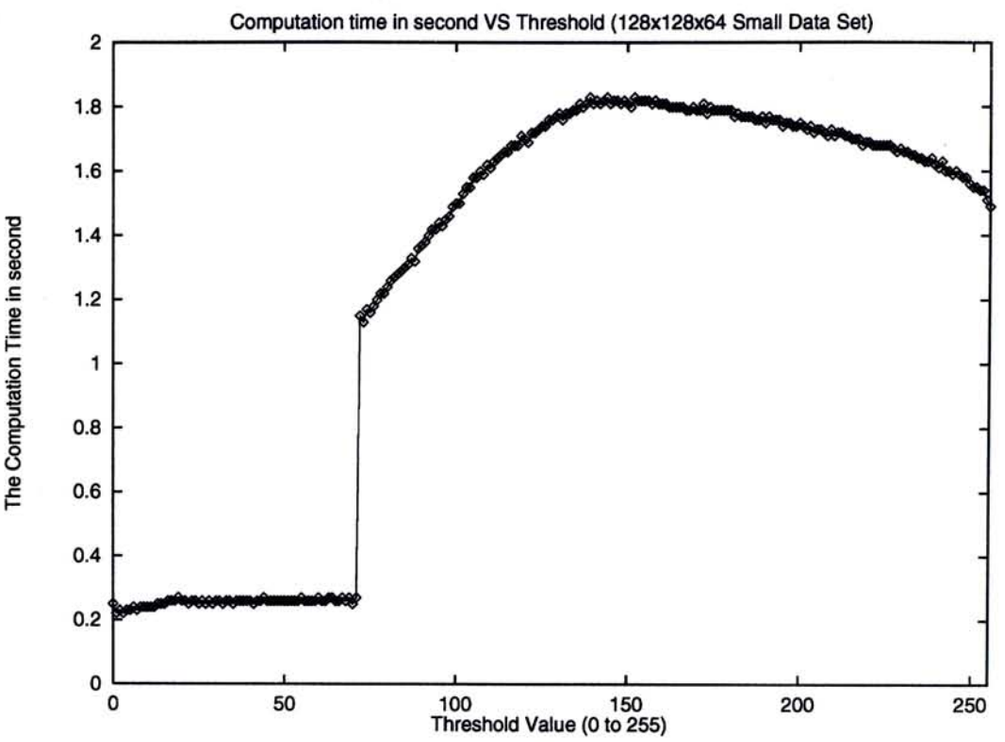



**Figure 7.4:** The Computation Time in Small Data Set (128x128x64)

**Figure 7.5:** The Reconstructed Bronchus (Small Data Set 128x128x64)

**Figure 7.6**: The Reconstructed Bronchus (Large Data Set 256x256x128)

image, the reconstructed bronchus (as shown in Fig. 7.5 and Fig. 7.6) of the smaller data set is less accurate and it has less tree branches. Limitation of resolution accounts for it. When the resolution decreases, it has the advantage of saving memory and execution time. On the other hand, its representative power is also decreased. The resultant medical visualization depends on the acquired image data.

To increase the efficiency of the extraction algorithm, we provide a much more accurate threshold value to the start point retrieval algorithm, or, if we know the start point, we can skip this process and directly proceed to the next procedure. Since the trachea has high contrast, we can figure out its location in the image without any difficulty. In addition, in the optimal threshold determination process, we can speed it up by giving a range of potential threshold values for the algorithm to perform testing instead of running with all possible threshold values.

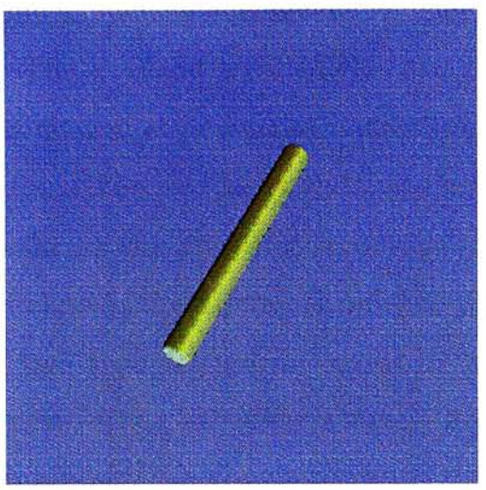
## 7.2 Experiment of Automatic Centerline Determination Algorithm

The automatic centerline determination algorithm is implemented and tested on real three dimensional CT lung data and five mathematical phantoms. Seven sets of lung airway tree segmented data are generated by the method mentioned in chapter 5 [26]. 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 grey-scale image, is prepared for comparison. Other six image sets contain around 185 images where each of them is a 256x256 8-bit grey-scale image.

Several mathematical phantoms are generated to evaluate the performance of centerline determination algoirthm in different typical circumstances. They are generated by using the known related centerline and radii information. The first phantom is a simple cylinder which is shown in Fig. 7.7(a) with a constant

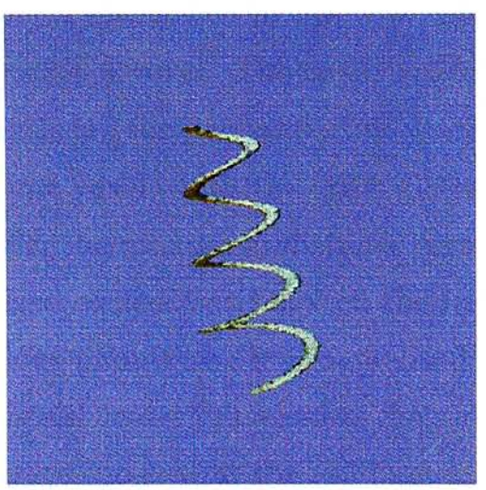
radius of 10 units. The basic unit of a three dimensional volume is a voxel. It is used to evaluate the simplest case. Fig. 7.7(b) shows a helix phantom with 20 units thick which is winding a 20-unit radius cylinder. Another helix phantom in Fig. 7.7(c) is with the same thickness and it is winding a prism having radii from 20 units to 38 units. The two helix phantoms are used to model the twisting case. Although lung airways do not have exteme twisting structures, they have curve structures. The fourth phantom in Fig. 7.7(d) is created by the centerline information derived from a real lung dataset. It has constant thickness of ten units. The final phantom, which simulates airway phantom, in Fig. 7.7(e) is generated and based on typical lung airway tree structure. It has airways from generation one to generation five. In real airway tree, the thickness of the airway is related to the corresponding generation number. When the generation number increases, the corresponding airways become thinner. To simulate this converging thickness property, different generation airways are assigned with various thickness. The radii are 10, 8, 6, 4 and 3 units for airways of generations 1, 2, 3, 4 and 5, respectively. Table 7.2 shows the detailed information.

All experiments are performed on Sun UltraSparc 5/270 workstation with Solaris 2.6 OS, 512MB RAM and 100Mbps network speed. Tables 7.1 and 7.2 present the running time of the algorithm for the data set and phantom data set correspondingly. The time is approximately linear to the number of object voxels. *distance_threshold* and *possible_distance_threshold* are set to be 3 and 26 respectively in the end points retrieval algorithm. User can alter these two distance thresholds whenever necessary. Accuracy evaluation of the algorithm is performed by using the mathematical phantoms with known centerline information. Table 7.3 shows the distance error results for various phantoms. Distance error is measured by the distance from a voxel in the generated centerline to the nearest voxel in the original centerline on the same slice in axial format. The distance unit is voxel. As the true centerline is non-discrete in value, it may cover the path between voxels. On the other hand, the generated

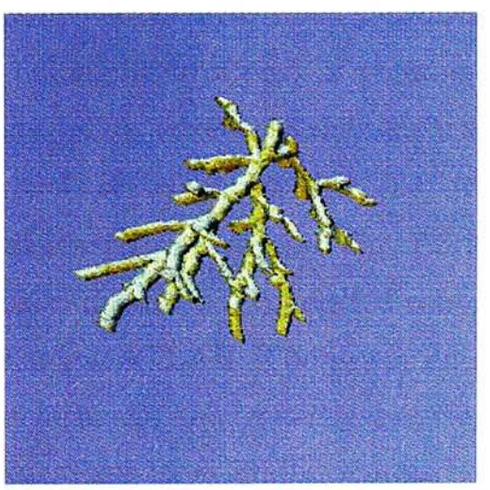


(a) Simple Cylinder Phantom

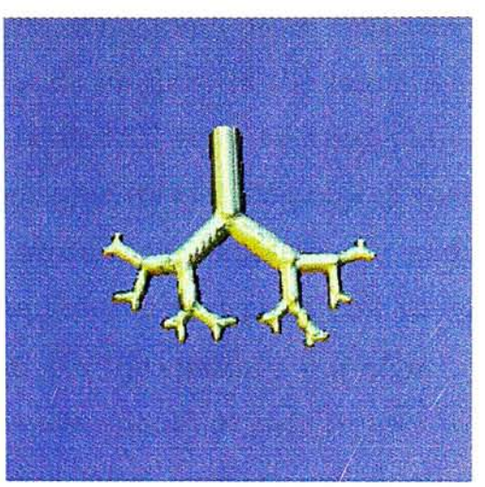
(b) Helix Phantom



(c) Helix with Vairous Radii Phantom



(d) Lung Airway Phantom



(e) Simulated Lung Airway Phantom

**Figure 7.7**: Mathematical Phantoms

centerline uses integer to represent the corresponding voxel coordinates within the volume, which is discrete. It can be expected that some errors are caused by it. The average error of the five phantoms is less than 1.1 voxel which indicates that the generated centerline does not critically depart from the real centerline. The two helix datasets have a maximum error of 5.34 voxels which may refer to the twisting structure. Because of the varying branching angles and thickness, the airway simulated phantom has maximum error of seven voxels. If two true centerline voxels in two continuous slices are departed from a few voxels, some inbetween voxels are included in the generated centerline so as to maintain the connectivity of the generated centerline. The newly added inbetween voxels have some voxels distance to the real centerline voxels. An example is illustrated in Fig. 7.9. It shows five slices with fixed y-coordinate. Each cell is a voxel. The color voxels are belonging to a part of volume. The red voxels are the true centerline voxels. The yellow voxels are the newly added centerline voxels with the error distance voxels stated inside the cells. In this example, the maximum distance error is at least three voxels. Fig. 7.8 is the visualized comparison between the real centerline and the generated centerline of various phantoms. It is visualized by using MATLAB. The green lines are the original centerline whereas the red lines form the new generated centerline. The figure shows that the red lines and green lines have similar shapes.

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 [2] which consists of 99054 object voxels with a 256x256x256 volume size requires 20 seconds to complete. In addition, the algorithm in Blezek [2] 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 it even does not require user to input any start point. In the case of similar object voxel number, our algorithm requires 15 seconds

| 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 7.1**: CPU Running Time of End Points Retrieval Algorithm, Distance Transform and Dijkstra Shortest Path Algorithm Applying to Real CT Datasets

| Phantom Data | Volume Size (voxels) | Object Voxels | Time(s) | Endpoint Voxels | Centerline Voxles |
|---|---|---|---|---|---|
| Cylinder | 256x256x185 | 59580 | 16.46 | 1 | 179 |
| Helix | 256x256x185 | 157303 | 24.1 | 1 | 414 |
| HelixSpecial | 256x256x185 | 217601 | 28.02 | 1 | 597 |
| Airway | 256x256x185 | 73778 | 17.3 | 41 | 1041 |
| AirwaySimulated | 200x200x185 | 46923 | 10.27 | 8 | 379 |

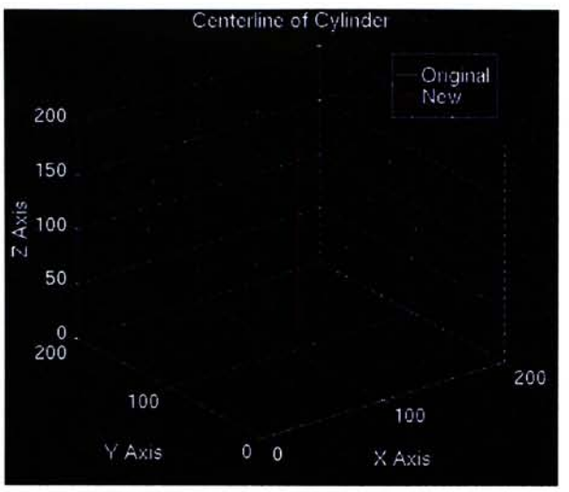
**Table 7.2**: CPU Running Time of Various Phantoms

which includes end points searching, distance transform and the Dijkstra's shortest path searching.

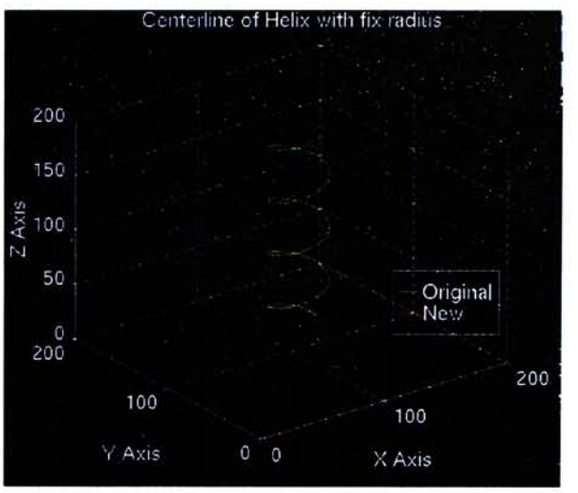
The extracted centerline is visualized with three dimensional texture mapping techniques [32] which is a direct data visualization technique that is similar to ray casting. Three dimensional textures are a logical extension of two dimensional textures. In three dimensional textures, texels become unit cubes in texel space. The three dimensional texture is used as a voxel cache, processes

| Phantom | Maximum Error (voxels) | Minimum Error (voxels) | Average Error (voxels) | Standard Deviation (voxels) |
|---|---|---|---|---|
| Cyliner | 0 | 0 | 0 | 0 |
| Helix | 5 | 0 | 0.67 | 0.66 |
| HelixSpecial | 5.34 | 0 | 1.02 | 0.85 |
| Airway | 5 | 0 | 0.85 | 0.79 |
| AirwaySimulated | 7 | 0 | 0.51 | 0.98 |

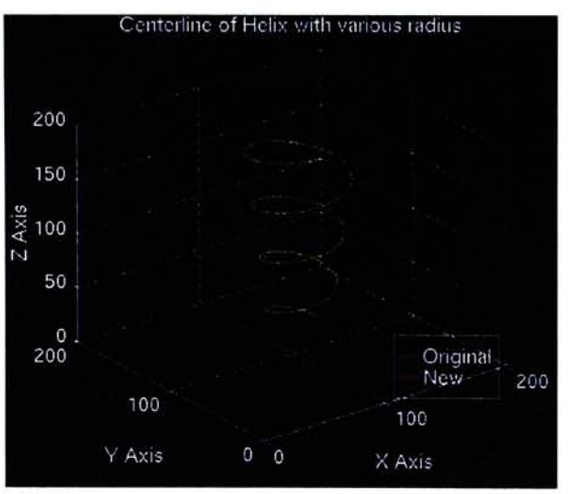
**Table 7.3**: Distance Error of Various Phantoms in voxels



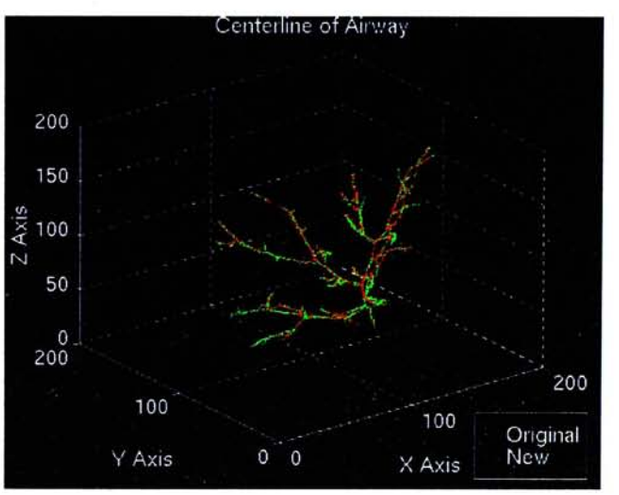

(a) Simple Cylinder Phantom



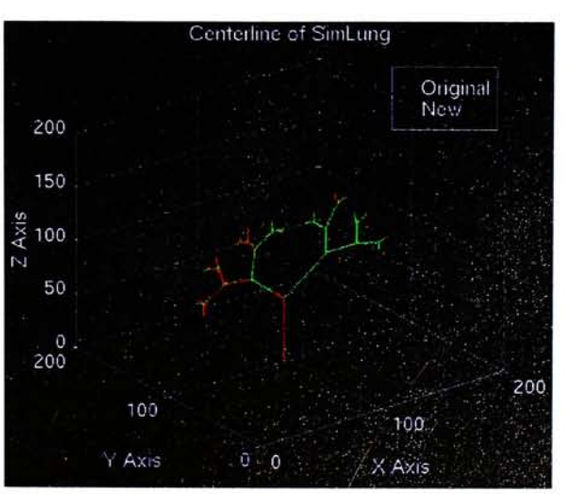

(b) Helix Phantom




(c) Helix with Vairous Radii Phantom




(d) Lung Airway Phantom




(e) Simulated Lung Airway Phantom

**Figure 7.8:** Visualization of Two Sets of Centerline in Various Phantoms

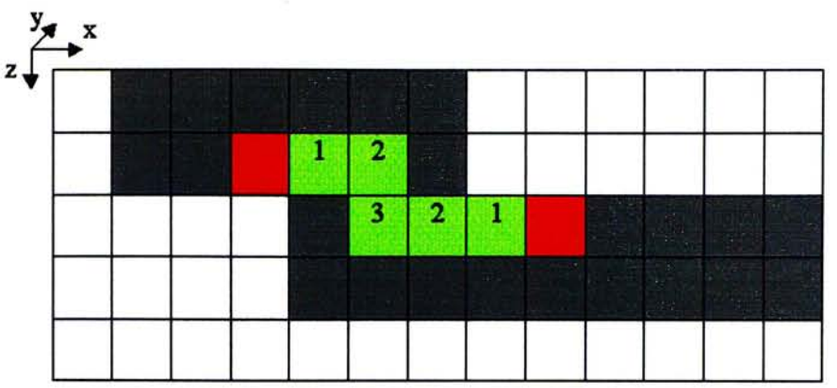
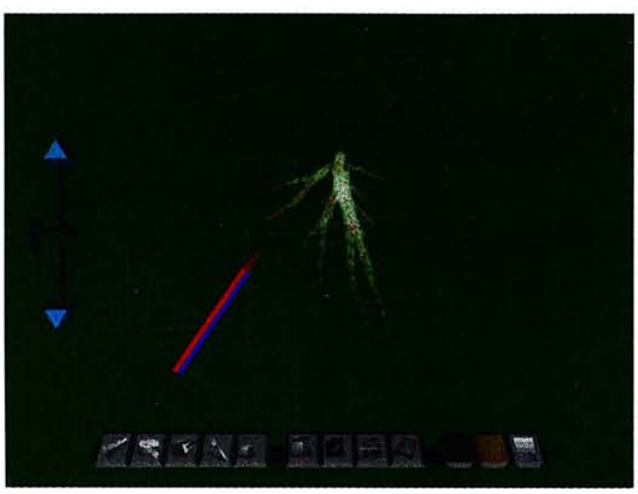
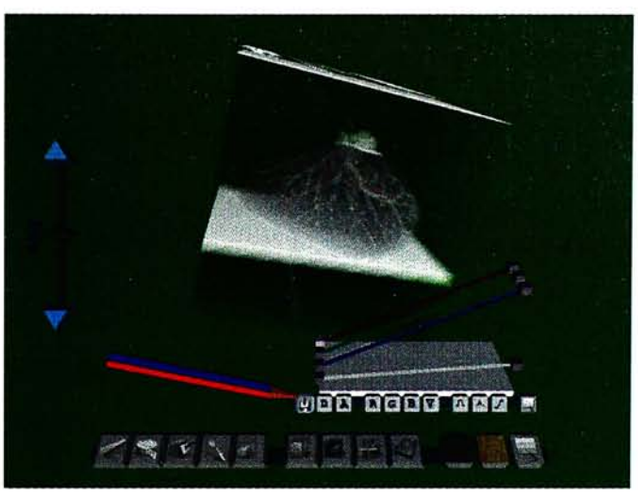



**Figure 7.9**: Explanation of distance error. The grey voxels are a part of volume. The red voxels stand for the true centerline voxels. The yellow voxels are the generated centerline voxels with distance error value inside the voxels.

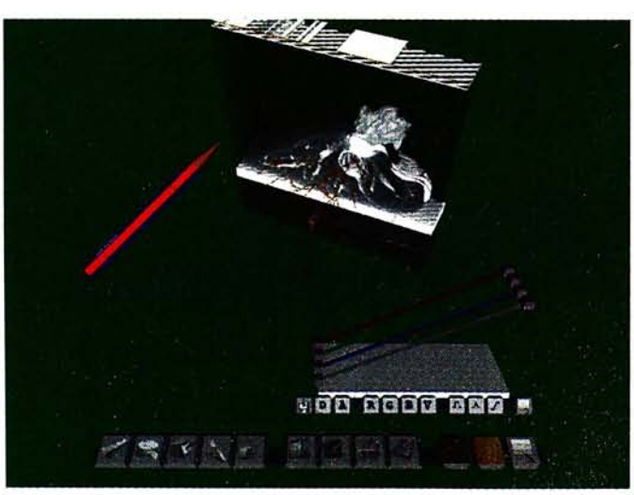
two dimensional layer each time by all rays simultaneously. Fig. 7.10 shows the images of the reconstructed centerlines. The centerlines are represented by red lines. Segmented lung data is visualized in Fig. 7.10(a) and (d). The lung data in Fig. 7.10(b) and (c) is not segmented. By setting proper colour map value, the non-airways lung volume (i.e. lung muscle volume) is viewed as transparent to allow the user to observe the airways more clearly. The centerlines are extracted satisfactorily.



(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

**Figure 7.10**: Various Visualization Results

# Chapter 8

# Conclusion

Using virtual reality becomes more popular in the medical imaging community. One major advantage of using virtual reality system is that it provides an initial assessment of the condition of patients. Furthermore, it takes less cost and risk. Bronchoscopy is a medical diagnosis for evaluating the endobronchial anatomy. Virtual bronchoscopy combines volumetric imaging and virtual reality technology. It decreases discomfort and inconvenience, considerably lower cost and risks. Simulation technology makes it possible for navigation to experience adverse scenarios without risk to human life or damage to expensive equipment. In addition, it is useful for training medical students or physicians to achieve better surgery skills.

We introduce our proposed procedure for automated extraction of bronchus area and determination of centerline of lung airway from three dimensional CT images for three dimensional virtual bronchoscopy. We apply the proposed extraction procedure which is based on genetic algorithm and region growing approaches to real three dimensional CT images. The experimental results show the ability of the algorithm to extract bronchus area and the algorithm works satisfactorily.

An automatic centerline determination algorithm from CT images for three dimensional virtual bronchoscopy is presented. The end points retrieval al-

gorithm 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. Our test cases include various CT image data sets. For a typical 256x256x180 segmented lung tree airway volume data, it requires around 15 seconds for the completion of the centerline determination procedure. To evaluate its accuracy, some artificial volume data with known centerline information are generated. The calculated centerline is very simliar to the real centerline and it proves the accuracy of our algorithm.

# Bibliography


[1] Daniel J. Blezek, Richard A. Robb, Charlene M. Prather, *Virtual Endoscopy vs. Real Endoscopy: A Comparative Evaluation*, SPIE Vol. 3031, 1997.

[2] 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 Multidimensional Images, San Diago, California, SPIE Vol. 3660, Feb 1999.

[3] Gunilla Borgefors, *Distance Tranformatoins in Arbitary Dimensions*, Computer Vision, Graphics, and Image Processing 27, 321-345, 1984.

[4] Gunilla Borgefors, *Distance Tranformations in Digital Images*, Computer Vision, Graphics, and Image Processing 34, 344-371, 1986.

[5] Martin L. Brady, Kenneth K. Jung et al., *Interactive Volume Navigation*, IEEE Transactions on Visualization and Computer Graphics, Vol 4., No. 3, July-Sept. 1998.

[6] Richard D. Bucholz, MD, FACS, *Advances in Computer Aided Surgery*, CAR'98.

[7] Thomos H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Introduction to Algorithms, The MIT Press, McGraw-Hill Book Company, 1990.

[8] Ney Derek et al, *Segmentation of Medical Images*, 3D Visualization in Medice, Siggraph 1992 Course Notes 34, pp. 38-48, July 1992.

[9] K.-H. Englmeier et al., *Virtual Bronchoscopy based on Spiral-CT Images*, SPIE Conference on Image Display, San Diego, California, SPIE Vol. 3335, pp. 427-438, Feb. 1998.

[10] Frasher and Pare, *Organ Physiology: Structure and Function of the Lung*, Second Edition, W. B. Saunders Company, 1977.

[11] Foley, Van Dam, Feiner, Hughes, Computer Graphics Principles and Practice, Second Edition in C, Addision Wesley, 1996.

[12] Y. Ge, David R. Stelts, and David J. Vining, *3D Skeleton for Virtual Colonoscopy*, Visualization in Biomedical Computing, LNCS 1131, pp. 449-454, Springer, 1996.

[13] Y. Ge, D. R. Stelts, X. Zha, J. Wang, and D. J. Vining, *Computing the Centeral Path of Colon Lumen in Helical CT Images*, Part of the SPIE Conference on Image Processing, pp. 702-713, SPIE Vol. 3338, 1998.

[14] Bernhard Geiger and Ron Kikinis, *Simulation of Endoscopy*, Lecture Notes in Computer Science 905, Computer Vision, Virtual Reality and Robotics in Medicine, First International Conference, CVRMed'95, Nice, France, Springer, pp. 277-281, April 1995.

[15] WeiXin Gong and Gilles Bertrand, *A Simple Parallel 3D Thinning Algorithm*, Pattern Recognition, 1990. Proceedings. 10th International Conference on Volume: i, Page(s): 188 -190 vol.1, 1990.

[16] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Addison Wesley, chaper 7, pp. 413-482, Sept. 1993.

[17] W.E. Higgins, E.A. Hoffman, G. McLennan, R. D. Swift, and J.M. Reinhardt, *Automatic image analysis for efficient virtual-bronchoscopic assessment*, Radiology, vol. 205 (P), pp. 358-358, Nov. 1997.

[18] W.E. Higgins, K. Ramaswamy, G. McLennan, E.A. Hoffman, and R. D. Swift, *A virtual-endoscopic system for interactive navigation and detailed quantitation*, invited paper, Radiographics, May 1998.

[19] L. Hong, S. Muraki, A. Kaufman, D. Bartz, and T. He, *Virtual Voyage: Interactive Navigation in the Human Colon*, Proc. ACM SIGGRAPH '97, pp. 27-34, 1997.

[20] Gordon W. Hunt, Paul F. Hemler, David J. Vining, *Automated Virtual Colonoscopy* SPIE Vol. 3031, 1997.

[21] K. Ikuta, M. Takeichi and T. Namiki, *Virtual Enodscope System with Force Sensation*, MICCAI' 98 Lecture Notes in Computer Science 1496, Springer, pp. 293-304, Oct. 1998.

[22] Fumikazu Iseki, Tsagaan Baigalmaa et al., *Extraction of 3D Tree Structure of Blood Vessels in Lung Area from Chest CT Images*, CAR'98, 1998.

[23] J. Ivins, J. Porrill, *Active Region Models for Segmenting Medical Images*, Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference Volume: 2 , Page(s): 227 -231 vol.2, 1994.

[24] M. Kass, A. Witkin, D. Terzopoulos, *Snakes: Active Contour Models*, First International Conference On Computer Vision, pp. 259-268,1987.

[25] A. Kriete, *Advances in Anatomy Embryology and Cell Biology Vol.145 Form and Function of Mammalian Lung: Analysis by Scientific Computing*, Springer, 1998.

[26] 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.

[27] Ta-Chih Lee, Rangasami L. Kashyap and Chong-Nam Chu, *Building Skeleton Models via 3D Medial Surface/Axis Thinning Algorithms*, CVGIP: Graphical Models and Image Processing, Vol. 56, No. 6, November, pp. 462-478, 1994.

[28] J.T.Littleton, M.L. Durizch, *Chest Atlas: Radiographically Correlated Thin-section Anatomy in Five Planes*, Springer-Verlag New York Inc., 1994.

[29] S. Lobregt, P.W. Verbeek and F. C. A. Groen, *Three-Dimensional Skeletonization: Principle and Algorithm*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 1, January 1980.

[30] Yoshitaka Masutani, Ken Masamune and Takeyoshi Dohi, *Region-Growing Based Feature Extraction Algorithm for Tree-Like Objects*, LNCS 1131 Höhne. Kikinis (Eds.), Visualization in Biomedical Computing, Springer, 1996.

[31] Yoshitaka Masutani, Thomas Schiemann and Kari-Heinz Höhne, *Vascular Shape Segmentation and Structure Extraction Using a Shape-Based Region-Growing Model*, LNCS 1498 Wells. Colchester Delp (Eds.), MICCAI'98 Medical Image Computing and Computer-Assisted Intervention, Springer, 1998.

[32] T. McReynolds et al., *Advanced Graphics Programming Techniques Using OpenGL*, SIGGRAPH 98 Course Notes 17, July 1998.

[33] 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.

[34] 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.

[35] 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.

[36] Chuzo Nagaishi, M.D., *Functional Anatomy and Histology of the Lung*, Igaku Shoin Ltd. Tokyo, 1971.

[37] C. Wayne Niblack, Phillip B. Gibbons and David W. Capson, *Generating Skeletons and Centerlines From the Distance Transfrom*, CVGIP: Graphical

Models and Image Processing, Vol. 54, No. 5, September, pp. 420-437, 1992.

[38] Ingela Nystrom, *Skeletonization Applied to Magnetic Resonance Angiography Images*, Part of the SPIE Conference on Image Processing, San Diego, California, SPIE Vol. 3338, Feb 1998.

[39] R. O'Tole, R. Player et al., *Assessing Skill and Learning in Surgeons and Medical Students Using a Force Feedback Surgical Simulator*, MICCAI' 98 Lecture Notes in Computer Science 1496, Springer, pp. 899-909, Oct. 1998.

[40] Chandrasekhar Pisupati, Lawrence Wolff et al., *A Central Axis Algorithm for 3D Bronchial Tree Structures*, IEEE International Symp. on Computer Vision, Miami, FL, pp. 259-264, 1995.

[41] Chandrasekhar Pisupati, Lawrence Wolff et al., *Segmentation of 3D Pulmonary Trees usng Mathematical Morphology*, International Symposium on Mathematical Morphology, Atlanta, GA, May 1996.

[42] Chandrasekhar Pisupati, Lawrence Wolff et al., *Tracking 3-D Pulmonary Tree Structures*, Proceedings of MMBIA'96, IEEE, 1996.

[43] Krishnan Ramaswamy and William E. Higgins, *Endoscopic Exploration and Measurement in 3D Radiological Images*, SPIE Conf. Medical Imaging 1996: Image Processing, vol. 2710, Newport Beach, CA, pp. 511-523, Feb. 1996.

[44] K. Ramaswamy and W.E. Higgins, *Coherence-Based Dynamic Navigation for Virtual Endoscopy*, SPIE Medical Imaging 1998: Image Display, Y. Kim and S.K. Mun, eds., SPIE Proceedings vol. 3335, 1998.

[45] Richard A. Robb, *Virtual Reality in Medicine and Biology*, Biomedical Imaging Recourse Mayo Foundation, Clinic Rochester, Minnesota

[46] Lawrence J. Rosenblum and Michael R. Macedonia, *Phantom-Based Haptic Interaction with Virtual Objects*, IEEE Computer Graphics and Applications, pp. 6-10, Sept.-Oct. 1997.

[47] Stuart J. Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, Prentice-Hall International, Inc., 1995.

[48] Yaseen Samara, Martin Fiebich et al., *Automated Centerline Tracking of the Human Colon*, Part of the SPIE Conference on Image Processing, San Diego, California, SPIE Vol. 3338, Feb 1998.

[49] Richard M. Satava et al., Future Technologies for Surgical Applications. Yim Book, Jan, 1998.

[50] Luis Serra, Tim Poston et al., *The Brain Bench Planner and Trainer for Minimal Access Surgery*, Proceedings of the ACM Symposium on Virtual Reality Software and Technolgy, pp. 191-192, July 1996.

[51] Milan Sonka, Christopher J. Wilbricht, Steven R. Fleagle et al., *Simultaneous Detection of Both Coronary Borders*, IEEE Transactions on Medical Imaging, Vol. 12, No. 3, September, 1993.

[52] Milan Sonka, Michael D, Winniford, Xiangmin Zhang and Steve M. Collins, *Lumen Centerline Detection in Complex Coronary Angiograms*, IEEE Transactions on Biomedical Engineering, Vol. 41, No. 6, June 1994.

[53] Erich Sorantin, B. Geiger, R. Fotter, Heinz Mayer, E. Eber, *Virtual Tracheobronchoscopy-How to Compute and How to Visualize?*, CAR'98.

[54] R. D. Swift, K. Ramaswamy, and W.E. Higgins, *Adaptive Axes-Generation Algorithm for 3D Tubular Structures*, 1997 IEEE Int. Conf. Image Processing, vol. II, Santa Barbara, CA, pp. 136-139, 26-29 Oct. 1997.

[55] R. D. Swift, W.E. Higgins, E.A. Hoffman, G. McLennan, and J.M. Reinhardt, *Automatic Axis Generation for 3D Virtual-Bronchoscopic Image Assessment*, SPIE Medical Imaging 1998: Physiology and Function from Multidimensional Images, E. Hoffman, ed., SPIE Proceedings vol. 3337, 1998.

[56] Ewald R. Weibel, *Design of Airways and Blood Vessels Considered as Branching Trees*, The Lung: Scienific Foundations, edited by R. G. Crystal, J. B. West et al., Raven Press, Ltd., New York, Chapter 4.2.1.1, 1991.

[57] Mark Allen Weiss, Data Structures & Algorithm Aanlysis in C++, Addison Wesley Books, Second Edition, 1999.

[58] Rüdiger Westermann, Thomas Ertl, *Efficiently Using Graphics Hardware in Volume Rendering Applicaitons*, ACM Siggraph 98 Conference Proceedings, 1998.

[59] Terry Yoo, *Segmentation and Classificaiton*, Siggraph 93 Course Notes 21, pp. 12-15 , Aug. 1993.

[60] Terry S. Yoo et al., *3D Visualization in Medicine 98*, Siggrah 98 Course Notes 23, 1998.