SKELETONIZATION OF TUBULAR OBJECTS USING SUPERQUADRIC ELLIPSOIDS

A Thesis in
Computer Science
by
Syed Majeed Amjad

© 2010 Syed Majeed Amjad

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2010
The thesis of Syed Majeed Amjad was reviewed and approved* by the following:

Thang N. Bui
Associate Professor of Computer Science
Chair, Mathematics and Computer Science Programs

Sukmoon Chang
Assistant Professor of Computer Science
Thesis Advisor

Qin Ding
Assistant Professor of Computer Science

Linda M. Null
Associate Professor of Computer Science
Graduate Coordinator

*Signatures are on file in the Graduate School.
Abstract

Anatomical structures contain various types of curvilinear or tube-like structures such as blood vessels and bronchial trees. Such tubular structures can be best represented by their skeletons. A skeleton of an object is the loci of center points of all maximally inscribed discs (or spheres in 3D). Skeletons reduce the complex 3D pixel-based representation of an object into a simpler 1D spatial line while preserving its boundary and region information. A skeleton should be thin, centered, and correctly connected and should preserve the topology of the object.

Many algorithms for skeletonization rely on the boundary information of an object to extract its skeletons. These methods are not feasible for medical images, where the boundary is not known a priori or the boundary is not clearly defined. For medical images, therefore, such methods require extensive pre-processing steps to identify the boundary of the objects of interest. Furthermore, these methods are sensitive to boundary noise. More recent approaches use ridge point detection for finding skeletons. Although these methods do not need boundary information they require complex grouping schemes to ensure the ridge points are correctly connected. These methods also require intensive user interaction, such as manual selection of seed points, which hinders the method from automation.

In this thesis, we propose a novel skeletonization algorithm using deformable models. The proposed algorithm requires minimal user interaction and no pre-processing steps to obtain boundary information. The algorithm generates thin, connected skeletons that preserve the topology of the object. Experiments on data show that the proposed algorithm extracts skeletons in sub-voxel accuracy, and results on clinical data also show accurately centered skeletons.
# Table of Contents

List of Figures .......................................................... vi

Acknowledgements ......................................................... viii

Chapter 1

Introduction ............................................................... 1

Chapter 2

Motivation ................................................................. 4

2.1 Existing Methods ................................................... 4

2.2 Skeletonization using Deformable Models ..................... 6

Chapter 3

Skeletonization with Deformable Models ......................... 8

3.1 Superquadric Ellipsoid ............................................ 9

3.2 Kinematics .......................................................... 11

3.3 Dynamics ........................................................... 13

Chapter 4

Forces for Deformable Models ........................................ 15

4.1 Local Smoothing .................................................... 15

4.2 The Edge Map and Force Computation ....................... 16

4.3 Gradient Vector Flow (GVF) .................................... 18
| Chapter 5 |
|-------------------------------|---|
| **Scan Conversion and PCA**   | 22 |
| 5.1 Scan Conversion Algorithm  | 22 |
| 5.2 Principal Component Analysis | 23 |

| Chapter 6 |
|-------------------------------|---|
| **Experimental Results**      | 27 |
| 6.1 Results for Simulated Data | 27 |
| 6.2 Results for Clinical Data  | 29 |

| Chapter 7 |
|-------------------------------|---|
| **Conclusion**                | 35 |

| References |
|-------------|---|
| References  | 37 |
List of Figures

1.1 Skeleton as the loci of centers of maximal discs. 2
1.2 Effects of boundary noise on skeletons. (a) Skeleton of a rectangle, (b) Skeleton of the same rectangle with boundary disturbances. 2

2.1 Distance map. (a) Original image, (b) Distance map for the original image. 5
2.2 An illustration of the first step of the skeletonization process. 7

3.1 Geometry of deformable superquadric. 9
3.2 Effect of the squareness parameter on Superquadric Ellipsoid. 11
3.3 Effect of the parameters on the reference shape of the proposed method. 12

4.1 Global smoothing v.s. local smoothing. (a) Original image, (b) Global Smoothing with Gaussian operator of size 3, (c) with Gaussian operator of size 11, (d) Local smoothing with different Gaussian operator size for each object. 17
4.2 Edge Map and Force computation. (a) Original image, (b) Smoothed image, (c) Edge map computed from the smooth image, and (d) the force vectors calculated from the edge map. 18
4.3 The GVF field. (a) The set of force vectors, (b) GVF field calculated from the force vectors in (a). 19
4.4 Model Deformation. (a) The model initialized in the 3D vessel, (b) The deformed model. 20
4.5 GVF fields for cylinder in Fig. 4.4. GVF field projected on (a) XY plane, (b) YZ plane, and (c) XZ plane. ................................. 21

5.1 Sign barriers. (a) Original image, (b) GVF field, (c) GVF field vectors projected along the horizontal axis, and (d) GVF field vectors projected along the vertical axis. ................................. 24

5.2 Ridge points on H-shaped object. (a) Distance map of the shape, (b) Sign barriers on $L_0$, and (c) Sign barriers on $L_{90}$. .............. 25

5.3 Ridge points and eigenvectors. (a) 3D vessel with bounding box and ridge points, (b) Eigenvectors $e_1$, $e_2$ and $e_3$. Vector $e_1$ is the principal component. ................................. 26

6.1 Experiment on simulated data. (a) Stack of images used to construct 3D cylinder, (b) Model tracing skeletal points, (c) Skeletal points (dots) and the known center line (solid line), and (d) Cross sections superimposed on the computed skeletal points. ..................... 28

6.2 Experiment on simulated data. (a) Stack of images used to construct 3D structure, (b) Actual 3D structure, (c) Skeletal points being traced by the deformable model, and (d) Skeletal points (dots) placed along the known center line (solid line). ...................... 30

6.3 Images from the dataset. .................................................. 31

6.4 Skeletons of 3D vessels. (a) Skeletal line traced by our algorithm, (b) Skeletal line for both branches, and (c) Cross sections superimposed on the skeletons. .................................................. 32

6.5 Skeletons of the vascular structure. (a) Tubular vessels from the left lung, (b) Skeletons computed by the proposed algorithm, and (c) Reconstructed vessels from (b). .................................................. 33
Acknowledgements

I would like to thank my advisor, Dr. Sukmoon Chang, for his valuable input and guidance in achieving the goals of this thesis. I am grateful to the members of the thesis committee: Dr. Thang N. Bui, Dr. Qin Ding, and Dr. Linda M. Null for reviewing this thesis.
Chapter 1

Introduction

In this thesis we are dealing with the problem of skeletonizing tubular objects in the human body. The human body contains various tubular elongated structures, such as blood vessels, lung bronchioles, and intestines. Skeletons are the best way of representing such elongated structures. They reduce the complex pixel-based 3D volume into a 1D spatial line in 3D space, while preserving the topology, region, and boundary information. This information can then be used in various image precessing tasks such as automated navigation, anatomical segmentation, visualization, and planning of surgical operations.

The notion of skeletons was first proposed by H. Blum [2]. The aim of skeletonization is to extract a boundary and region-based shape representation that preserves the topology of an object. A skeleton is defined to be the loci of the center points of all the maximally inscribed discs (or spheres in 3D). Fig. 1.1 shows the skeleton formed by the loci of the centers of maximally inscribed discs for a rectangle. Formally, a skeleton is defined as follows:

**Definition 1.1:** Given an object, a skeleton is the set of pixels (2D) or voxels (3D) that lie equidistant from at least two nearest boundary points of that object.

The definition of a skeleton can also be illustrated by the grass-fire analogy: the boundary of the object of interest is set on fire, which propagates inwards at a
constant speed. Then, the skeleton is the loci of points where the fire fronts meet and quench each other out.

The property of a skeleton is that it should be thin. It should contain much fewer points than the original object (i.e., an object reduction). A skeleton should also preserve region and boundary information and should conform to the topology of the object.

The weakness of the approach for skeletonization as described above is that, in order to find such skeletons, we need to have a priori information about the boundary. In addition, such an approach is very sensitive to boundary disturbances. Fig. 1.2(a) shows the skeleton of a rectangle and (b) shows how it changes completely after introducing some boundary disturbances.

The approach for skeletonization described above is also infeasible for use in medical images, since medical images are noisy and the boundaries of the objects are not clearly defined. Therefore, in order to find skeletons from medical images, pre-processing or segmentation is required to extract boundary information first.

We propose a novel method for skeletonization to find the skeletons for 3D tubular objects (3D vessels) present in medical images without requiring any pre-processing steps to obtain boundary information. The proposed algorithm uses
Deformable models to skeletonize a 3D tubular vessel. Deformable models are physics-based models that deform when forces are applied on them [18]. In the proposed method, we use a Gradient Vector Flow (GVF) field as the deformation forces [21, 22]. Once the deformable model fits to the boundary of the vessel, the center of this model represents the skeletal point of the 3D vessel at that location.

In order to trace the skeletal line of the vessel, we have to deform the model at successive locations along the elongated direction of the 3D vessel. The new location for deformation is determined by moving the model in the elongated direction of the vessel by a small distance. We will show that the direction of the vessel can be estimated by the scan-conversion algorithm for ridge point detection [3] followed by the Principal Component Analysis (PCA) [9]. Once it is re-initialized at the new location, the model is deformed to obtain the next skeletal point. This process is repeated until the desired skeletal line of the vessel is found.

We applied our method to simulated data sets for arbitrarily oriented 3D cylinders. The average mean square error for the resulting skeletal lines was 0.00629, which illustrates that our algorithm computes skeletons in sub-voxel accuracy. We also present promising results of the proposed algorithm applied to pulmonary vessels from clinical chest CT images.
Chapter 2

Motivation

In this chapter, we review existing skeletonization algorithms and examine their strengths and weaknesses. We then present our algorithm and discuss how it addresses the problems of existing algorithms.

2.1 Existing Methods

The skeleton is a shape representation based on both the boundary and the region information of an object. A skeleton has the following properties:

- Lying along the center with respect to the object boundary
- Preserving boundary and region information of the object
- Preserving topology of the object

Boundary peeling and distance maps are two widely used skeletonization algorithms [13]. The boundary peeling method iteratively removes the simple points on the boundary of an object until the skeleton is reached. Simple points are those points that can be removed without affecting the topology of the object. However, identifying the simple points is not trivial and the resulting skeletons depend on the criteria for determining the simple points [13]. In addition to these problems, the peeling method is sensitive to boundary noise and the skeleton is not always thin. In the distance map method, a distance map is constructed from the 3D volume [3]. A distance map for a 3D volume is computed by assigning each point inside an object the distance from this point to the nearest boundary
2 Motivation

Figure 2.1. Distance map. (a) Original image, (b) Distance map for the original image.

point. Local maxima are then detected from these distance maps and connected to form the skeletal line. Fig. 2.1 shows the distance map approach [23]. Fig. 2.1(a) shows the original image and (b) shows the distance map for that image. These methods cannot, however, be applied directly to images where the boundary is not clearly defined. Therefore, these methods require pre-processing steps to obtain the boundary information before running the algorithm.

In medical images, the objects of interest appear as relatively brighter (or darker) regions and the intensity ridges of these regions tend to lie towards the center of the object. Therefore, intensity ridges have been used as an alternative way of approximating the location of skeletons [6]. Algorithms based on intensity ridges can be classified into two categories. The first category consists of finding the ridge points in a 3D volume and connecting these points in a post processing step [7]. Since this approach uses local criteria for ridge point identifications, it can generate false positives for ridge points. Thus, in order to obtain connected skeletons, this approach needs to use elaborate grouping schemes [11]. The second approach constructs skeletons by searching for ridge points around a neighborhood of a known ridge point [1]. This method generates thin, connected skeletons without any complex post-processing steps. However, the search criteria for the ridge points can be computationally expensive. It also requires a set of criteria or heuristics to handle special cases such as branching points and end-points. Furthermore, this approach requires the manual selection of seed points which hinders the method from automation.

In this thesis, we propose a new skeletonization method that has the following advantages over the existing methods:
2 Motivation

- No pre-processing or segmentation, i.e., no need for a priori boundary information
- Minimal user interaction
- Insensitivity to boundary noise

2.2 Skeletonization using Deformable Models

The proposed algorithm uses deformable models to find the skeletal points and to trace the skeletons of a 3D tubular object. The idea is to fit a deformable model to the boundary at a certain location in the 3D tubular object. Once the model deforms and fits the boundary of the 3D tubular object, the center of the fitted sphere model is the skeletal point of the 3D object at that location.

In order to find the complete skeleton, the model must be deformed at successive locations along the elongated direction of the tubular object. Successive deformation locations are computed by the Principal Component Analysis (PCA) on the set of ridge points found by applying the scan-conversion algorithm on the Gradient Vector Flow (GVF) field [9, 3]. At this new location, the model is initialized and deformed again to obtain the next skeletal point. This process is repeated until the desired skeletal line for the tubular object is computed.

Our method requires minimum user interaction. A user provides an initial point, the approximate diameter of the vessel at the point, and the initial direction vector. Fig. 2.2 shows the initialization of the proposed method. The cylinder, the sphere, and the arrow in the figure represent the 3D tubular object, the deformable model (initialized at the user provided initial point), and the initial direction vector, respectively. With these parameters, the proposed method first constructs a region of interest around the initial point. In the figure, the box around the initial point represents the region of interest constructed by the proposed method.

The forces for deformation are then computed inside the region of interest as the Gradient Vector Flow (GVF) field [21, 22]. These forces are applied on the deformable model to make it deform and fit to the boundary of the vessel at that location. After deformation, the center of the fitted model represents the skeletal point of the tubular object at that location.
By using deformable models in our algorithm, we do not need any a priori boundary information. The deformable model interacts with the vessel based on the forces that are calculated directly from the grayscale medical images. The proposed algorithm also guarantees thinness of the skeleton, since the model traces the skeletons along the center of the object. This method is semi-automatic, with the user having to provide only the initial point, the initial diameter of the vessel, and the initial direction vector. The rest of the algorithm executes without any further user interaction.

We present the results of our method applied on simulated data as well as clinical data. The simulated data is comprised of grayscale images for various 3D cylinders. Results on the simulated dataset show that our method can compute the skeletons in sub-voxel accuracy. The clinical dataset comprised of CT scans of the human lung. Preliminary results on this dataset also show that the generated skeletal lines are well aligned with the center of the 3D vessel.
Chapter 3

Skeletonization with Deformable Models

The goal of our algorithm is to skeletonize a 3D tubular object. The problems with existing algorithms for skeletonization are that they are sensitive to boundary noise and require a priori boundary information as well as intensive user interaction. To overcome these problems, we use deformable models.

Deformable models are a powerful model-based image analysis technique [12, 18]. Such models have been used to identify and segment objects present in images [10]. Deformable models are physics-based models that deform at the application of forces that can be calculated directly from the image data. For this reason, deformable models are especially useful for medical images, where the object boundaries are not clearly defined. Furthermore, our problem consists of finding the skeletons for elongated objects that have almost circular cross sections, for example, blood vessels, arteries, and bronchioles. Deformable models can easily fit to the contour of such a cross section, based on the forces calculated directly from the image data, without the need for a priori boundary information.

In our algorithm, we use a Superquadric Ellipsoid as a reference shape for the deformable model for modeling the medical images [19]. A reference shape for a deformable model is the original model with global deformation. The global deformation determines how the reference shape changes according to the underlying equations for deformation. This chapter describes the geometry of the superquadric ellipsoid and the deformable model framework.
3.1 Superquadric Ellipsoid

Geometrically, deformable models are closed surfaces in space whose intrinsic (material) coordinates $u=(u,v)$ are defined on domain $Ω$ [19]. The positions of points on the model relative to an inertial frame of reference $Φ$ in space are given parametrically by a vector-valued, time-varying function of $u$:

$$x(u, t) = (x_1(u, t), x_2(u, t), x_3(u, t))^T$$  \hspace{1cm} (3.1)

where $T$ is the transpose operator. These positions can be expressed in cartesian coordinates as:

$$x = c + Rp$$  \hspace{1cm} (3.2)

where $c$ is a vector defining the origin of $Φ$ at the center of the model in the cartesian coordinate system, and the orientation of $Φ$ is given by the rotation matrix $R$. Thus, $p$ denotes the canonical positions of points on the model relative to the model frame [19]. We further express $p$ as the sum of a reference shape $s$ and a displacement function $d$:

$$p = s + d$$  \hspace{1cm} (3.3)
Fig. 3.1 illustrates the geometry of the deformable model.

The formulation for any reference shape can be given as a parameterized function of \( u \). We use a superquadric ellipsoid as the reference shape for the proposed method:

\[
    s = a_0 \begin{pmatrix} a_1 C_u^\epsilon_1 C_v^\epsilon_2 \\ a_2 C_u^\epsilon_1 S_v^\epsilon_2 \\ a_3 S_u^\epsilon_1 \\ \end{pmatrix}
\] (3.4)

where \(-\pi/2 \leq u \leq \pi/2\), \(-\pi \leq v < \pi\), and 

\[ S_u^\epsilon = \text{sgn}(\sin u)|\sin u|^\epsilon, \quad C_u^\epsilon = \text{sgn}(\cos u)|\cos u|^\epsilon, \quad \text{and similarly for } C_v^\epsilon \text{ and } S_v^\epsilon. \]

Here, \( a_0 \geq 0 \) is a scale parameter, \( 0 \leq a_1, a_2, a_3 \leq 1 \) are aspect ratio parameters, and \( \epsilon_1, \epsilon_2 \geq 0 \) are squareness parameters [12, 18]. We collect the parameters into the parameter vector:

\[
    q_s = (a_0, a_1, a_2, a_3, \epsilon_1, \epsilon_2).
\] (3.5)

These are the parameters that will determine the shape of the model as it deforms and fits into the tubular object. Fig. 3.2 shows the effect of the squareness parameters \( \epsilon_1 \) and \( \epsilon_2 \). Fig. 3.2(a) is a superquadric ellipsoid with \( \epsilon_1 = 1 \) and \( \epsilon_2 = 1 \). Fig. 3.2(b), (c), and (d) show the effects of changing these squareness parameters.

In this thesis, the objects of our interest are tubular objects, such as blood vessels, that are present in medical images. Since these objects have an oblong appearance with almost circular cross sections, we do not require local deformations \( d \), and the squareness parameters \( \epsilon_1 \) and \( \epsilon_2 \). Therefore Eq. (3.5) becomes:

\[
    q_s = (a_0, a_1, a_2, a_3).
\] (3.6)

Fig. 3.3 shows the different shapes achieved by changing these parameters. In the next section we will show the equations that govern how these parameters are updated based on the external forces.
3.2 Kinematics

This section presents the kinematic formulation of how a point on the deformable model is updated based on the different parameters of the superquadric ellipsoid. These parameters are also called generalized coordinates. This kinematic formulation leads to the Jacobian Matrix $L$ which allows the transformation of 3D vectors into $q$-dimensional vectors, where $q$ is the number of degrees of freedom of the deformable model.
Figure 3.3. Effect of the parameters on the reference shape of the proposed method.

The velocity of points on the model from Eq. (3.2) is given by

\[ \dot{x} = \dot{c} + \dot{R}p + R\dot{p} \]

\[ = \dot{c} + B\dot{\theta} + R\dot{s} \quad (3.7) \]

where \( \theta = (\theta_1, \theta_2, \theta_3)^T \) is the vector of rotational coordinates of the model, and \( B = [\partial (R\dot{p})/\partial \theta_1, R\dot{p}/\partial \theta_2, R\dot{p}/\partial \theta_3] \). Furthermore,

\[ \dot{s} = \left[ \frac{\partial s}{\partial q_s} \right] \dot{q}_s = J\dot{q}_s \quad (3.8) \]
where, $J$ is the Jacobian of the superquadric ellipsoid with respect to $q_s$. Thus, Eq. (3.7) can be written as:

$$\dot{x} = [I \ B \ R \ J] \dot{q} = L \dot{q}$$  \hspace{1cm} (3.9)

Here,

$$q^T = (q_c^T, q_\theta^T, q_s^T)$$  \hspace{1cm} (3.10)

with $q_c = c$ and $q_\theta = \theta$, and $q_s$ serving as the vector of generalized coordinates for the dynamic model.

The matrix $L$ is a Jacobian matrix that maps the generalized coordinates, i.e. the parameters $q$, into 3D vectors. The details of the derivation of the Jacobian matrix $J$ along with the matrices $I$, $B$, and $R$ can be found in [19].

### 3.3 Dynamics

When fitting the model to 3D data, our goal is to update $q$, which is the vector of the degrees of freedom of the model. The components $q_c$ and $q_\theta$ are the global rigid motion coordinates and $q_s$ are the global deformation coordinates. Through the apparatus of Lagrangian dynamics, we arrive at a set of the equations of motion governing the behavior of our model under the action of externally applied forces [19]. The Lagrange equations of motion take the form

$$M \ddot{q} + C \dot{q} + K q = g_q + f_q$$  \hspace{1cm} (3.11)

where $M$, $C$, and $K$ are the mass, damping, and stiffness matrices, respectively, $g_q$ are inertial forces arising from the dynamic coupling between the local and global degrees of freedom, and $f_q$ are the generalized external forces associated with the degrees of freedom $q$ of the model. A detailed derivation of this equation can also be found in [19].

For most applications in computer image analysis involving the fitting of a hollow deformable shell to static data, we may simplify the underlying motion equations for computational efficiency. In our application, we want the hollow shell to fit the data and achieve equilibrium as soon as the forces balance each
other out. For such a problem, it makes sense to simplify the motion equations by
setting the mass density $M$ to zero. Setting the mass density $M$ to zero causes
the inertial forces to disappear. Eq. (3.11), therefore, can be simplified as:

\[ C \dot{q} K q = f_q \]  

(3.12)

Therefore, our model relies on the external forces to guide its deformation and the
system comes to rest as soon as all the forces equal out. The formulation of our
model yields numerically stable equations of motion that may be integrated forward
through time using explicit procedures. For fast interactive response, we employ
a first-order Euler method to solve Eq. (3.12). The Euler procedure updates the
degrees of freedom $q$ of the model at time $t + \Delta t$ according to the formula [19]:

\[ q^{(t+\Delta t)} = q^{(t)} + \Delta t(C^{(t)})^{-1}(f_q^{(t)} - K q^{(t)}) \]  

(3.13)

The forces for the deformable model, $f_q$, are calculated directly from the
grayscale image data. These forces then act on the degrees of freedom of the
model and make it deform. The following chapter describes how we calculate the
forces for the proposed skeletonization method.
Chapter 4

Forces for Deformable Models

For the superquadric ellipsoid to deform, we need to apply appropriate forces to the model. These forces are 3D vectors calculated directly from the image data. The following sections describe the computer vision techniques used to derive the forces for our model. We start by explaining local smoothing, followed by the Edge Map, and then the Gradient Vector Flow (GVF) field that computes the force vectors from the smoothed image data.

4.1 Local Smoothing

In order to find the forces from the image data, the images must first be smoothed. This is necessary because there is noise present in the medical images that needs to be smoothed. This section describes the smoothing operations used on the image data.

The Gaussian operator [4] is a convolution operator used to smooth images and remove any noise that is prevalent in medical images. The 3D Gaussian distribution has the form:

\[
G_{\sigma} = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x^2 + y^2 + z^2)}{2\sigma^2}}
\]  

(4.1)

where \( \sigma \) is the standard deviation of the distribution. The 3D smoothing operator can be represented as follows:

\[
I'(x,y,z) = G_{\sigma} * I(x,y,z)
\]  

(4.2)
where $I'(x,y,z)$ is the smoothed 3D volume, $I(x,y,z)$ the original 3D image volume, $\ast$ the convolution operator, and $G_{\sigma}$ the 3D Gaussian smoothing operator.

The size of the Gaussian operator $\sigma$, determines the degree of smoothing, i.e., the larger the size of the Gaussian operator, the more the image will be smoothed. The size of the Gaussian operator should not be constant since there are objects of varying sizes present in medical images. If the size of the Gaussian operator is constant (i.e., global smoothing), then it is possible that regions of interest that are very small might be smoothed more than necessary and the regions that are large might not be smoothed at all. In order to overcome this problem, we have introduced local smoothing in our algorithm. Fig. 4.1 shows the effect of global smoothing versus local smoothing on 2D images. Fig. 4.1(a) is the original image, and (b) and (c) are the smoothed images using global smoothing with various sizes of the Gaussian operator. In Fig. 4.1(b), the size of the Gaussian operator is 3, that is equal to the width of the bottom line. We can see that the bottom line is properly smoothed but the top-most line is incorrectly smoothed. Similarly, in Fig. 4.1(c), the size of the Gaussian operator is 11, that is equal to the width of the top-most line. We can see again that the top-most line is properly smoothed whereas the bottom line is smoothed too much. The optimal size of the Gaussian operator is the width of the object being smoothed. In local smoothing, the size of the Gaussian operator is set to be equal to the width of the object. This way local smoothing can overcome the problems of too much smoothing or too little smoothing. The effect of local smoothing is illustrated in Fig. 4.1(d). Each line is smoothed appropriately based on the width of the object.

The proposed algorithm uses local smoothing by setting the size of the Gaussian operator to be equal to the diameter of the vessel. This is done by creating a region of interest around the deformable model and smoothing inside this region. Therefore, we can perform local smoothing with the best Gaussian operator size.

4.2 The Edge Map and Force Computation

To deform and fit the model to the 3D vessel, we need the forces that will attract the model to the boundary of the vessel. To compute such forces, we first compute
an edge map, $E(x, y, z)$, as the following energy function [10]:

$$E(x, y, z) = |\nabla I(x, y, z)|^2$$

(4.3)

where $I(x, y, z)$ is the image and $\nabla$ is the gradient operator. Note the edge map has the property that the energy is larger near the object edges as illustrated in Fig. 4.2(c). This figure is calculated by applying the edge map on Fig. 4.2(b) that is calculated by smoothing the original image shown in Fig. 4.2(a).

The deformation forces are computed as:

$$f(x, y, z) = \nabla E(x, y, z)$$

(4.4)

Fig. 4.2(d) shows an example of the forces generated by this method in 2D. The vector field in the figure has forces pointing towards the edges. These forces attract the model toward the edges of the object. Note, however, that the forces are mostly around the boundary and, thus, have a very small capture range. Therefore, in order for this force to attract the model to the boundary of the object, the model must be initialized very close to the boundary of the object. One solution to this
Figure 4.2. Edge Map and Force computation. (a) Original image, (b) Smoothed image, (c) Edge map computed from the smooth image, and (d) the force vectors calculated from the edge map.

The basic idea of GVF is to propagate the vectors all over the image space so that the deformable model has a greater chance of being attracted towards the edges. The GVF field in 3D is defined to be the vector field:

\[
\mathbf{v}(x,y,z) = (u(x,y,z), v(x,y,z), w(x,y,z))
\]  

(4.5)
It minimizes the energy function:

\[ \varepsilon = \int \int \int \mu |\nabla v|^2 + |\nabla f|^2 |v - \nabla f|^2 \, dx \, dy \, dz \]  

(4.6)

This formulation follows a simple idea, that of propagating the existing forces to the places where there are less or no forces. In particular, we see that when \( |\nabla f| \) is small, the energy is dominated by the partial derivatives of the vector field yielding a smooth field propagated over a region with lesser forces. On the other hand, when \( |\nabla f| \) is large, the second term dominates the integrand, and is minimized by setting \( v = \nabla f \). The parameter \( \mu \) is a regularization parameter governing the tradeoff between the first term and the second term. This parameter should be set according to the amount of noise present in the image [21].

Using the calculus of variation, it can be shown that the GVF field can be found by solving the following Euler equations [21]:

\begin{align*}
\mu \nabla^2 u - (u - f_x)(f_x^2 + f_y^2 + f_z^2) &= 0 \\
\mu \nabla^2 v - (v - f_y)(f_x^2 + f_y^2 + f_z^2) &= 0 \\
\mu \nabla^2 w - (w - f_z)(f_x^2 + f_y^2 + f_z^2) &= 0
\end{align*}

(4.7)  

(4.8)  

(4.9)

Fig. 4.3(b) shows the GVF field propagated from the gradient vectors in Fig. 4.3(a). As can be seen, the vectors around the edge have been propagated all over the im-
age space. This means that the deformable model has a much greater chance of being attracted towards the edges, even if it is initialized away from the boundary of the vessel. Fig. 4.4 shows the deformable model fitting inside a cylinder. Fig. 4.4(a) shows the initialized model and (b) illustrates the deformed model after fitting to the boundary of the cylinder. The model fits correctly to the cylinder because the forces point towards the boundary of the cylinder as illustrated in Fig. 4.5, in which the GVF forces are projected along three orthogonal planes. We can see that these forces correctly point towards the boundary of the cylinder.

Until now, we have shown a method that determines one skeletal point. In order to find the skeletal line of a vessel, we have to move the model along the vessel and deform it at successive locations in order to trace the entire skeletal line. The next chapter describes how we have used the scan-conversion algorithm [3] and Principal Component Analysis [9] to calculate the next approximate location for the model to deform.
Figure 4.5. GVF fields for cylinder in Fig. 4.4. GVF field projected on (a) XY plane, (b) YZ plane, and (c) XZ plane.
Chapter 5

Scan Conversion and PCA

This chapter describes the process by which we determine the successive locations for model deformation to trace the skeletal line of a vessel. The process which we use in our algorithm combines the scan-conversion algorithm and principal component analysis (PCA). The PCA method finds the vector along which there is greatest variance in the data (called the principal component). For PCA to find this vector, we need a set of data points that lie around the center of the 3D vessel. This set of data points (or ridge points) is found by a method called Scan-Conversion algorithm. The following sections describe this process in more detail.

5.1 Scan Conversion Algorithm

In grayscale images including medical images, the objects of interest usually appear as relatively brighter (or darker) regions based on their intensity. These regions have intensity ridges that lie around the center of the objects. The scan-conversion algorithm detects these ridge points based on the GVF field of the objects.

In our implementation, the GVF vectors point away from the ridge points toward object boundary. Therefore, for a point to be on a ridge, there must be a sign barrier in some direction, i.e., on a line passing through that point [3]. Consider a line \((L_{\theta})\) with an arbitrary orientation \(\theta\) and three contiguous points \((p_{-1}, p_0, p_{+1})\) on \(L_{\theta}\). If \(p_0\) is on a ridge, the gradient vectors at \(p_{+1}\) and \(p_{-1}\) must point away from \(p_0\), forming a sign barrier [3]. Fig. 5.1(b) illustrates an example
of a sign barrier around a ridge of an object in Fig. 5.1(a). Note that these sign barriers are not easily identifiable at some part of the object. We can enhance these sign barriers by projecting the GVF field vectors onto a line \( L_\theta \). This is illustrated in Fig. 5.1(c), where \( \theta = 0 \) (horizontal line), and Fig. 5.1(d), where \( \theta = 90 \) (vertical line).

If a ridge intersects \( L_\theta \), the ridge must form a sign barrier on \( L_\theta \) and, thus, is guaranteed to be detected. However, if a ridge is nearly parallel to \( L_\theta \) and does not intersect it, the ridge may or may not produce a sign barrier on \( L_\theta \). Therefore, we use another line \( L'_\theta = L_\theta + 90 \), so that such a ridge point can be detected [3]. This is shown in Fig. 5.2. Fig. 5.2(a) is the distance map of an \( H \)-shaped object. Fig. 5.2(b) and (c) show the sign barriers on \( L_0 \) and \( L_{90} \) respectively. Note that the horizontal ridge in the middle of the object is not present in (b), while the vertical ridges are not present in (c). The horizontal ridge, as expected, is detected in (c) with \( L_{90} \) while the vertical ridges are detected in (b) with \( L_0 \).

The proposed algorithm is dealing with 3D tubular objects. We learn from [3] that we need three orthogonal lines, \( L_x \), \( L_y \) and \( L_z \), parallel to the \( x \), \( y \) and \( z \) axes, respectively, in order to find the ridge points for a 3D tubular object. This is illustrated in Fig. 5.3. We have applied the scan-conversion method on a 3D cylinder and Fig. 5.3(a) shows the ridge points that have been detected.

Now that we have these ridge points, we need to find the vector approximating the direction along which to move our deformable model. This is done through the principal component analysis as described in the next section.

## 5.2 Principal Component Analysis

Principal Component Analysis (PCA) is a technique used in statistical analysis to simplify a data set by reducing its multidimensionality [9]. We use PCA on the set of the detected ridge points of the 3D vessel to find the elongated direction of the vessel.

Given a set of data points (i.e., the detected ridge points), PCA computes eigenvalues \( \lambda_1 , \lambda_2 , \lambda_3 \), and the corresponding eigenvectors \( e_1 , e_2 , e_3 \). The eigenvector corresponding to the largest eigenvalue shows the direction along which the data set has the greatest variance [9]. In other words, assuming, \( \lambda_1 > \lambda_2 > \lambda_3 \),
Figure 5.1. Sign barriers. (a) Original image, (b) GVF field, (c) GVF field vectors projected along the horizontal axis, and (d) GVF field vectors projected along the vertical axis.
the eigenvector $\mathbf{e}_1$ is the vector pointing in the direction of the greatest variance in the data set as shown in Fig. 5.3(b). In our algorithm, we approximate the next point for the model deformation by moving a small distance, $\delta$, along $\mathbf{e}_1$. 

**Figure 5.2.** Ridge points on H-shaped object. (a) Distance map of the shape, (b) Sign barriers on $L_0$, and (c) Sign barriers on $L_{90}$. 

5 Scan Conversion and PCA
Figure 5.3. Ridge points and eigenvectors. (a) 3D vessel with bounding box and ridge points, (b) Eigenvectors $e_1$, $e_2$ and $e_3$. Vector $e_1$ is the principal component.
Chapter 6

Experimental Results

We applied the proposed algorithm on simulated datasets of arbitrarily oriented cylinders with known center points. We also applied the proposed algorithm on clinical datasets of the human lung. In the deformation process of our algorithm we have a time-step value $\Delta t$ that determines how we update the parameters of the model. There is a time-step/accuracy trade-off in our algorithm. The lower the time-step the more accurate the result of the deformation process, however it is more time consuming. Similarly, the higher the time-step the less accurate the deformation process, but it is less time consuming. We determine a time-step that requires few iterations and gives us an accurate result from the deformation process. The time-step value, $\Delta t$, used in our algorithm is 0.001.

In order to determine the stopping condition of the deformation process we have used a threshold value. If the change in our updated parameters as compared to the previous parameters falls below this threshold then the deformation stops. The threshold value used in our algorithm is 0.0001. On average our algorithm took 32 iterations at every deformation step. The following sections describe the results achieved by the proposed algorithm.

6.1 Results for Simulated Data

To verify the correctness of the proposed method, we first applied the method to simulated data with known skeletons. The simulated data contained 10 cylinders in 3D space. Fig. 6.1 illustrates the results of the proposed method applied to
one arbitrarily oriented cylinder. Fig. 6.1(a) shows the stack of images used to construct the 3D cylinder. Fig. 6.1(b) shows the 3D cylinder with the known center line and the model tracing the skeletal points. Fig. 6.1(c) shows the computed skeletal points as well as the known center line and (d) shows the perpendicular cross sections of the cylinder superimposed on the computed skeletal points. As can be seen in Fig. 6.1(c) and (d), the computed skeleton is accurately aligned with the known centers of the cylinder.

Table 6.1 compares the skeletal points computed by the proposed algorithm and
Table 6.1. Simulated data results

<table>
<thead>
<tr>
<th>Points</th>
<th>Actual Point</th>
<th>Generated Point</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>1</td>
<td>15.26</td>
<td>15.26</td>
</tr>
<tr>
<td>2</td>
<td>18.15</td>
<td>18.15</td>
</tr>
<tr>
<td>4</td>
<td>23.91</td>
<td>23.91</td>
</tr>
<tr>
<td>6</td>
<td>29.69</td>
<td>29.69</td>
</tr>
<tr>
<td>7</td>
<td>32.58</td>
<td>32.58</td>
</tr>
<tr>
<td>8</td>
<td>35.46</td>
<td>35.46</td>
</tr>
<tr>
<td>9</td>
<td>38.35</td>
<td>38.35</td>
</tr>
<tr>
<td>10</td>
<td>41.24</td>
<td>41.24</td>
</tr>
</tbody>
</table>

the actual center points of the cylinder. The average mean square error for the skeletal points of all the 3D cylinders was 0.00535, which shows that our algorithm extracts the skeletal points in sub-voxel accuracy.

The proposed algorithm was also applied on a complex set of cylinders with tapering and bending. This dataset is illustrated in Fig. 6.2. Fig. 6.2(a) shows the stack of images used to construct this dataset. Fig. 6.2(b) shows the 3D volume constructed from these images. Fig. 6.2(c) shows the deformable model tracing the skeletal points of this structure, and (d) shows the computed skeletal points placed against the known center line of the structure. As can be seen the computed skeletal points lie on the known center points of the structure.

Furthermore, table 6.2 compares the skeletal points computed by the proposed algorithm and the actual center points. The mean square error for these points was 0.00536. This shows that our algorithm was able to achieve sub-voxel accuracy for this 3D structure as well.

### 6.2 Results for Clinical Data

We also applied the proposed algorithm to a clinical chest CT to skeletonize pulmonary vessels. The original data set consists of 196 CT scan images of a human lung, three of which are shown in Fig. 6.3. These images are stacked up to form the 3D volume for the lung. For our experiment, we extracted a 3D volume of the
left lung with the dimensions of $256 \times 256 \times 196$ voxels.

Fig. 6.4(a) shows a view of a 3D vessel from the data set that is being skeletonized by the deformable model. The deformable model moves along the lower branch as it determines the skeletal points. The dots in this figure represent the computed skeletal points of the 3D vessel. In order to skeletonize the next branch in the 3D vessel, another starting point was initialized at the branching point. The algorithm then proceeded to skeletonize the adjoining branch as shown in Fig. 6.4(b). The experiments were performed on a Dell Inspiron computer with
### Table 6.2. Simulated data results

<table>
<thead>
<tr>
<th>Points</th>
<th>Actual Point</th>
<th>Generated Point</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>1</td>
<td>343.725</td>
<td>477.462</td>
</tr>
<tr>
<td>2</td>
<td>350.247</td>
<td>458.34</td>
</tr>
<tr>
<td>3</td>
<td>349.17</td>
<td>441.139</td>
</tr>
<tr>
<td>4</td>
<td>339.064</td>
<td>427.043</td>
</tr>
<tr>
<td>5</td>
<td>321.949</td>
<td>416.054</td>
</tr>
<tr>
<td>6</td>
<td>302.519</td>
<td>406.985</td>
</tr>
<tr>
<td>7</td>
<td>286.356</td>
<td>397.915</td>
</tr>
<tr>
<td>8</td>
<td>277.791</td>
<td>386.926</td>
</tr>
<tr>
<td>9</td>
<td>278.254</td>
<td>372.83</td>
</tr>
<tr>
<td>10</td>
<td>285.728</td>
<td>355.629</td>
</tr>
<tr>
<td>11</td>
<td>295.516</td>
<td>336.508</td>
</tr>
<tr>
<td>12</td>
<td>302.037</td>
<td>317.386</td>
</tr>
<tr>
<td>13</td>
<td>300.961</td>
<td>300.185</td>
</tr>
<tr>
<td>14</td>
<td>290.855</td>
<td>286.09</td>
</tr>
<tr>
<td>15</td>
<td>273.74</td>
<td>275.1</td>
</tr>
<tr>
<td>16</td>
<td>254.31</td>
<td>266.031</td>
</tr>
<tr>
<td>17</td>
<td>238.147</td>
<td>256.961</td>
</tr>
<tr>
<td>18</td>
<td>25.4695</td>
<td>398.246</td>
</tr>
<tr>
<td>19</td>
<td>41.9057</td>
<td>393.469</td>
</tr>
<tr>
<td>20</td>
<td>65.195</td>
<td>395.545</td>
</tr>
</tbody>
</table>

### Figure 6.3. Images from the dataset.
Figure 6.4. Skeletons of 3D vessels. (a) Skeletal line traced by our algorithm, (b) Skeletal line for both branches, and (c) Cross sections superimposed on the skeletons.
6 Experimental Results

Figure 6.5. Skeletons of the vascular structure. (a) Tubular vessels from the left lung, (b) Skeletons computed by the proposed algorithm, and (c) Reconstructed vessels from (b).

1.6 GHz Intel pentium 4 processor and 512 MB of memory. This process took approximately 115 seconds. In Fig. 6.4(c), the cross sectional images of the vessel were superimposed onto the computed skeletons. Through visual inspection of the cross sections, we can see that the computed skeleton accurately passes through the center of the 3D vessel. Fig. 6.5 shows a result of the proposed algorithm applied
to a more complex vascular structure. Fig. 6.5(a) shows a volume of interest for the network structure of the vessels of the left lung, and (b) shows the computed skeletons using the proposed algorithm. Fig. 6.5(c) shows the reconstructed vessels from the computed skeletons.
Chapter 7

Conclusion

In this thesis, we proposed a novel skeletonization algorithm that uses deformable models to find the skeletons of 3D tubular objects. The proposed algorithm overcomes many problems with existing skeletonization methods.

Most existing algorithms require a priori boundary information to compute skeletons. These algorithms are not directly applicable to medical images, where the boundary is not clearly defined. Our algorithm requires no a priori boundary information. It computes the skeletons of 3D tubular objects using deformable models directly from the medical images without any pre-processing.

More recent approaches to skeletonization use ridge point detection to extract skeletons. However, since this approach uses local criteria to find the ridge points, they frequently generate false positives. Thus, these approaches require elaborate grouping schemes as a post-processing step to obtain thin skeletons that are correctly connected. On the other hand, the proposed method generates skeletal lines that are intrinsically thin and correctly connected without any post-processing, since our method traces skeletal points along the elongated direction of the 3D object.

Finally, many of the existing algorithms require intensive user interaction, such as selecting seed points, which hinders the methods from automation. The proposed algorithm, however, requires minimal user interaction. The user provides the initial point, the direction vector, and the approximate diameter of the 3D vessel at the initial point. The rest of the algorithm works automatically.

We presented the promising results of the algorithm on the simulated data of 3D
Conclusion

cylinders and clinical data of a human lung. Results on several simulated datasets showed that the skeletal points computed from our algorithm are aligned with the actual skeletal points for the 3D tubular objects, with an average mean square error of only 0.00535. The low mean square error suggests that our algorithm is able to extract skeletons in sub-voxel accuracy. Preliminary results on clinical data showed, from visual inspection of the cross sections, that there is good alignment between the detected points and the center-points of the vessels.

Possible future work may include detecting the branching points of a vessel automatically. We can initialize two different deformable models at this location in order to skeletonize both the branches. This will lead to an algorithm that can fully automate the extraction of skeletons for a complex structure of tubular objects.
References


