

**The Pennsylvania State University
The Graduate School
College of Engineering**

FAST SKELETONIZATION OF BLOOD VESSELS

A Thesis in
Computer Science
by
Aaron A. Croasmun

© 2012 Aaron A. Croasmun

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2012

The thesis of Aaron A. Croasmun was reviewed and approved* by the following:

Sukmoon Chang
Associate Professor of Computer Science
Thesis Advisor

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

Linda M. Null
Associate Professor of Computer Science
Associate Chair, Mathematics and Computer Science Programs

Jeremy J. Blum
Assistant Professor of Computer Science

Omar El Ariss
Assistant Professor of Computer Science

*Signatures are on file in the Graduate School.

Abstract

The study of the morphological and rheological behaviors of intramural vessels plays a critical role in various clinical applications such as surgical planning and radiotherapy. To better understand the rheological behavior of vascular structures in relation to the network morphology, we must obtain the concrete measurements of the morphometric parameters of the vascular networks under various conditions. Morphometric parameters of the networks include vessel diameter, branching points, branch end points, branch length, and branching angles. Because of the complexity of blood vessel morphology, however, it is difficult to obtain accurate measurements. In this thesis, we present a novel and efficient method for skeletonization of intramural vessel networks. The proposed method automatically skeletonizes the vascular network in a given image and constructs a graph structure that represents the branching structures of the network. Since the method processes a given image as a whole, the multiple vascular networks present in the image are automatically detected and skeletonized simultaneously. Moreover, since the skeletons are represented as graph structures, various morphometric parameters can be obtained automatically. We present the promising results of the proposed method applied to the complex structure of retinal vessel networks.

Table of Contents

List of Figures	v
List of Tables	vi
Acknowledgments	vii
Chapter 1	
Introduction	1
Chapter 2	
Previous Work	5
2.1 Introduction	5
2.2 Analytical Computation	5
2.3 Direct Thinning	8
2.4 Ridge-Following	9
Chapter 3	
Fast Skeletonization of Blood Vessels	12
3.1 Enhancing Vessels	12
3.2 Extracting Ridge Points	14
3.3 Building Skeletons	20
3.4 Smoothing Skeletons	23
Chapter 4	
Results	28
Chapter 5	
Conclusion	34
5.1 Summary	34
5.2 Future Work	35
Bibliography	37

List of Figures

1.1	MRA image of a human retina	2
1.2	MRA image of a human retina with hemorrhages	3
1.3	Skeletons of an elongated object	4
2.1	Effect of boundary noise on skeletons	6
2.2	Wavefront propagation from an object boundary	7
2.3	Skeleton extracted from projected lines	7
2.4	Voronoi polygons	8
2.5	Direct thinning applied to elongated object	9
2.6	Distance map representation of an image	9
2.7	Ordered Region Growing	10
2.8	3D representation of a vessel crawler	11
3.1	Intensity cross sections of vessels in an MRA image	13
3.2	Matched filter kernels	14
3.3	A matched filter image	15
3.4	Distance map of H-shaped object	16
3.5	Intensity profile surface	17
3.6	Sign change patterns of the projected gradient vectors on a scanline	17
3.7	Scanline result	19
3.8	Intensity cross section of noise and vessel	20
3.9	Ratio filter result	21
3.10	Pruning spurious branches	24
3.11	Result of the smoothing method	26
4.1	Proposed method results for STARE images	29
4.2	Results comparison of other methods and proposed method for a STARE image .	30
4.3	Results comparison of other methods and proposed method for a STARE image .	31
4.4	Proposed method result for abnormal STARE image	32
4.5	Results comparison for an abnormal STARE image	33

List of Tables

4.1	Proposed method's average processing time in milliseconds	29
-----	---	----

Acknowledgments

I wish to thank my thesis advisor, Dr. Sukmoon Chang, for his guidance and limitless tolerance of me. Without him, neither this thesis nor my graduate career would exist.

Words are a poor substitute for the gratitude I feel for all the professors under whom I have studied these past years; Dr. Thang Bui, whose mind-stretching lectures showed me the incredible depth of computer science; Dr. Linda Null, whose enthusiasm inspired me to learn and sparked, in me, an enthusiasm for computer science I may have otherwise never had; Dr. Jeremy Blum - while I only had the privilege of being apart of one of his classes, I cannot forget his guidance and willingness to help, both of which I am grateful for; Dr. Omar El Ariss, who was a member of my thesis committee; and again my thanks to Dr. Sukmoon Chang, who went the extra mile (or three) to further my academic career, who put up with my endless procrastination, torrent of questions and self-doubts, and who showed me that research is fascinating!

I am deeply grateful, as well, to Mr. Paul Eppley, Dr. Qin Ding, Dr. Pavel Naumov, Dr. Clifford Wagner, Ms. Julia Lobur, and last, but not least, Mr. John Furlas whose teachings I carry with me to this day.

I also owe countless thanks to my friends at Penn State; Jon, Andrew, Bill, Galin, Doug, Keith, and Nick. We spent endless hours studying, gaming, studying, laughing, studying, snacking, studying, and studying. I would not have made it this far without them. They are brilliant, no question about it.

I would also like to thank my coworkers, who supported me and graciously tolerated my need for an unusual work schedule.

Without the support and encouragement of my friends and family I would not be where I'm at today. So, to my mom, my aunt Shirley and uncle Jim, Brant, Chris, Christine, Peter, Heather, Mike, Shannon, Kris, Jamie, Brian, Steve, and Tony, thank you, all.

Finally, I owe a special thanks to my wife, Kate. She supported me, encouraged me, and waited patiently (more patiently than me, I might add) for me to complete this degree. She is the foundation of my life and though I cannot express it eloquently enough, my gratitude for her is infinite.

Introduction

The study of morphological and rheological behaviors of intramural vessels and vascular networks plays a critical role in various clinical applications. Cardiologists study them to accurately diagnose cardiovascular diseases and to help properly plan surgical procedures [1]. Ophthalmologists examine vessels in retinal images to diagnose diabetes, hypertension, and arteriosclerosis [2]. Researchers use them to help create theories of how vascular networks develop [3]. To do such tasks, clinicians and researchers must measure the morphological parameters of vascular networks. These morphological parameters include branch width, branch length, branch curvature, branching points (bifurcation points), branching angles, and branch end points. To automatically measure these parameters, vascular networks must be detected and represented in a shape representation that facilitates such analyses.

Detecting vascular networks in medical images is difficult for many reasons. Vascular networks exhibit a complexity created by bifurcation points, varying branch lengths, and erratic curvatures. The Magnetic Resonance Angiography (MRA) image in Figure 1.1, from the publicly available “Structured Analysis of the Retina” (STARE) database [4], illustrates the seemingly chaotic complexity of vascular networks (the dark lines) in a human retina. In addition to the network complexity, notice in the figure that some vessels cross each other creating false-positive bifurcation points. MRA images with low resolution also make it difficult to detect vessels since random background noise can easily be mistaken for real vessels. Moreover, detecting very fine vessels can be hard as they can be mistaken for noise. In addition, pathology such as hemorrhages and microaneurysms may cause difficulty in the automatic detection of vessels [5]. For example, the dark blotches in Figure 1.2 [4] are hemorrhages that make it difficult to determine the paths of intersecting vessels, even for a human observer.

To facilitate the automatic measurement and analysis of the morphological parameters, the vascular networks detected must have a proper shape representation. Biological objects of elongated shapes can be best represented by their centerlines (or skeletons) along their elongated direction as illustrated in Figure 1.3(a). The process of computing centerlines, called skeletoniza-



Figure 1.1. MRA image of a human retina

tion, of elongated objects was first proposed by Blum [6]. In his paper, Blum defined skeletons as the centers of maximally inscribed discs in an object that are not completely contained within another disc as illustrated in Figure 1.3(b). Skeletons must satisfy three properties. First, they must be centered along the elongated objects. Second, they must guarantee thinness; in the case of digital images, they are no more than one pixel wide. Finally, skeletons must preserve connectivity of objects. Any reasonable skeletonization method should maintain these properties in its results.

Throughout the years, many skeletonization methods have been proposed. However, most existing methods share several problems that prevent them from being fully automated. These problems include user intervention [7], a priori knowledge of vessel boundaries [8], and post processing to measure parameters [9]. In this thesis, we present a novel and efficient method for automatic skeletonization of vascular networks. The proposed method automatically detects the centerlines of vessels in a given image and constructs a graph structure that represents the vascular networks in the discrete image space. To obtain a smooth skeleton in the continuous



Figure 1.2. MRA image of a human retina with hemorrhages

space, we apply active contour models to the discrete skeleton graph. Our method has several advantages over existing methods. It requires neither human intervention nor a priori knowledge of vessel boundaries. Since the proposed method processes a given image as a whole, multiple vascular networks present in an image are automatically detected and skeletonized simultaneously. Moreover, since the skeletons are represented as graph structures, various morphological parameters such as branch lengths and branching points can be automatically obtained.

We begin this thesis with a discussion of existing skeletonization methods and their limitations in Chapter 2. We then present our method in detail in Chapter 3. Chapter 4 presents the very promising results of our method applied to the complex vascular networks in MRA images of human retina. Finally, we present concluding remarks and future work in Chapter 5.

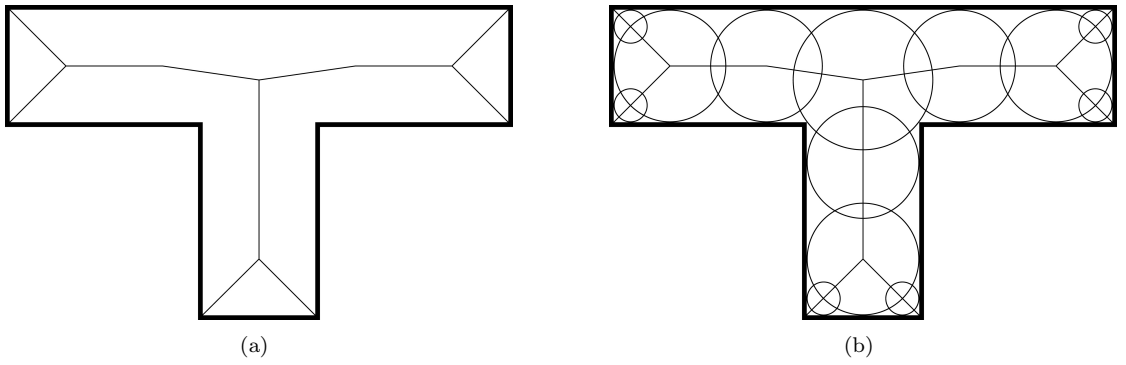


Figure 1.3. Skeletons of an elongated object. (a) Skeleton formed by centerlines of an object. (b) Collection of centers of maximally inscribed discs connected to create a skeleton within an object

Previous Work

2.1 Introduction

Blum defined the skeleton of an object as a collection of the center points of maximally inscribed discs where no disc is contained completely within another. Blum also described the process of computing skeletons as an analogy to “grassfire propagation” [6]. The basic idea behind grassfire propagation is to ignite a fire along an object boundary. The firefronts burn towards the center of the object at a constant speed. Wherever the firefronts meet, they are extinguished. These extinguishing points are equivalent to the centers of maximally inscribed discs. The grassfire propagation method has three main properties: a point inside an object has a value 0 or 1 (burned or unburned), a point may be ignited from an external source (an object boundary) or an adjacent burning point, and a burned point cannot be re-ignited to prevent reburning. There are some drawbacks to this method. It requires a priori knowledge of the object boundaries to provide initial points for ignition. Additionally, small noise on the object boundaries can cause spurious centerlines where firefronts meet very early. This problem is illustrated in Figure 2.1. The arrow in the figure points to a small noise on the object boundary, which creates a false-positive centerline. Note that to provide visibility of the skeleton not all maximally inscribed discs are shown in the figure.

Since Blum’s definition, many skeletonization methods have been developed to extract and represent the centerlines of objects, each with their own strengths and weaknesses. These methods generally fall into three categories: analytical computation, direct thinning, and ridge-following. In this chapter, we review methods from each category and discuss their strengths and weaknesses.

2.2 Analytical Computation

Analytical computation is a skeletonization method that uses geometry to represent an object as a set of polygons. These polygons are then used to calculate the object’s centerline.

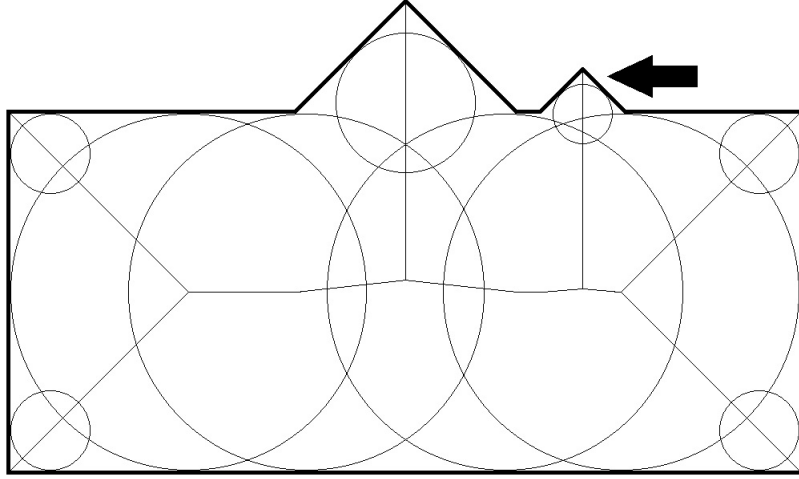


Figure 2.1. Effect of boundary noise on skeletons. The arrow points to object boundary noise which creates a false-positive centerline

Montanari introduced a skeletonization method using analytical computation and wavefront propagation [10]. The basic premise of wavefront propagation is to generate waves within an object and record where the waves meet. Waves start at an object boundary and propagate inwards as illustrated in Figure 2.2. Centerline points are those points at which two or more waves intersect. Objects in this method are expressed as closed connected polygons and wavefronts begin at the polygon boundaries. In order to propagate the waves, this method calculates successive wavefronts using Huygen’s principal. Given any wavefront W , a successive wavefront can be calculated as the envelope or circular wavefront of radius r generated at every point of W [10]. There are various disadvantages with this method regarding the concavity and convexity of the polygon vertices. As waves propagate inwards from boundary noise in the form of a concave vertex, they begin intersecting immediately. This often causes false-positive centerlines. On the other hand, a wave propagating inwards from a convex vertex becomes an arc and does not precisely represent the vertex from which it originated. Thus, any wavefront intersections at these arcs are imprecise and may not represent the true centerline.

Another method of analytical computation was developed by Martinez et al. [11]. Similar to the previous method, this method creates a polygonal representation of an object. Instead of using wavefront propagation, this method projects lines inwards from the polygon vertices until they intersect with a polygon edge on the object boundary. The vertex from which a projected line starts and the edge it intersects are termed a “matching pair.” If the projected line is valid, based on certain criteria, the mid-point of the projected line is considered a point of the object centerline. A finite number of lines are projected from each vertex and at different angles. Once all of the lines have been projected, the midpoints of valid lines are connected to create a centerline of the object. Figure 2.3 shows projected lines as dotted arrows. Note that only some of the projected lines are shown. The dashed lines in the center of the object represents

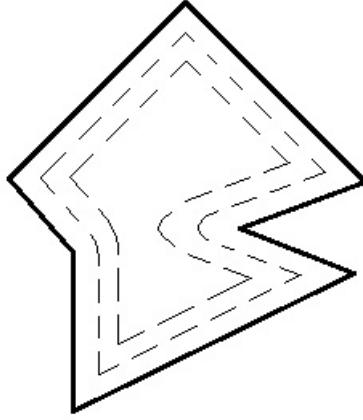


Figure 2.2. Wavefront propagation from an object boundary

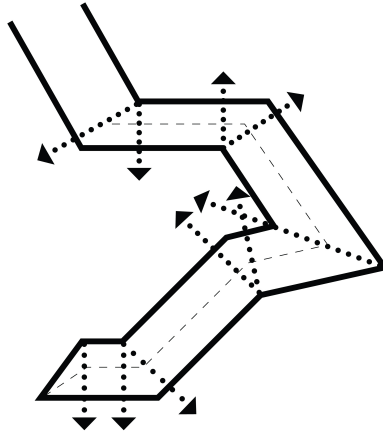


Figure 2.3. Skeleton extracted from projected lines. Note only some projected lines are shown for clarity

the centerlines of the object. A drawback to this method is that the selection of matching pairs is computationally expensive and new pairs must be selected many times while processing an image. Another drawback to this method is that center points can be identified “out of order.” When these “out of order” points are connected to form a centerline, small loops can occur.

Ogniewicz et al. presented a method utilizing Voronoi diagrams to calculate the centerlines of objects [9]. A Voronoi diagram is illustrated in Figure 2.4(a). In this figure, Voronoi polygons were created around a random set of points on a plane. A feature of Voronoi polygons is that their edges are equidistant from two or more points in the set of points. The authors exploited this feature in order to calculate an object’s centerlines. This method selects a set of points on

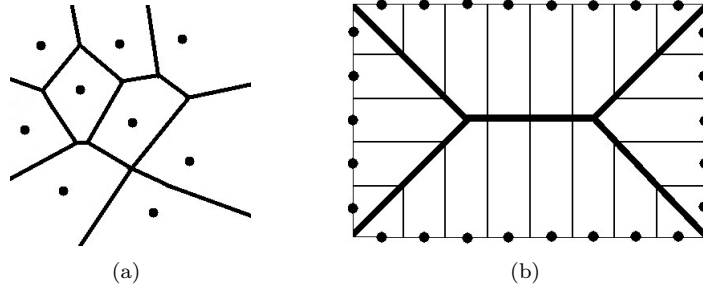


Figure 2.4. Voronoi polygons. (a) Voronoi polygons created around a set of points in a plane. (b) Voronoi polygons exposing the skeleton of a rectangle

the boundaries of objects and generates Voronoi polygons around them. The result is a Voronoi diagram where at least one edge of each Voronoi polygon constitutes a part of the centerline of an object. This is illustrated in Figure 2.4(b). In the figure, the centerline is represented by the bold line in the middle, and the dots are the set of points on the object boundary. One of the weaknesses of this method, which can easily be seen in Figure 2.4(b), is that most of the Voronoi polygon edges have to be pruned and selecting the correct edges for pruning is difficult. Aggressive pruning could remove sections of the centerlines. Another weakness is exposed when choosing a set of points on the boundary. Too many points result in unnecessary skeleton branches while too few points result in an imprecise skeleton.

There are two intrinsic weaknesses inherent to analytical computation methods. First, creating a polygonal representation of an object requires a priori knowledge of the object boundaries. Second, biological objects tend to be difficult to represent as polygons when accuracy is important [8].

2.3 Direct Thinning

Unlike analytical computation methods, which calculate object centerlines from polygon representations of the object, the direct thinning method iteratively peels off the contours of an object until only a thin centerline is left. Direct thinning methods are applied to topological surfaces of images, such as distance maps; a record of each pixel's distance from the nearest object boundary pixel. The basic idea of direct thinning is illustrated in Figure 2.5. In this figure, an elongated object is represented as a distance map. Each successive image shows how boundary pixels are peeled off until finally only a centerline is left. For clarity, all the images display the original distance value for each pixel.

Arcelli et al. introduced a direct thinning method called Topological Thinning [12]. This method operates on a distance map and, based on certain criteria, iteratively removes pixels from the object boundary until a stopping criterion is met. This method looks for a “thin set” of pixels in the image. The thin set contains only those pixels that have at least one neighboring pixel not in the set. Thus, the thin set is at most two pixels wide [12]. However, while pixels in

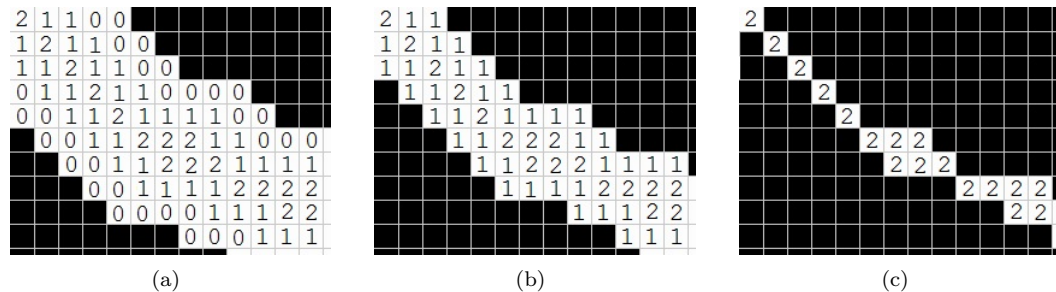


Figure 2.5. Sequential view of direct thinning applied to a distance map of an elongated object

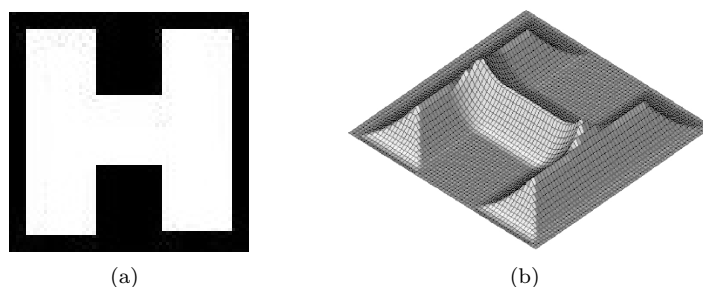


Figure 2.6. Distance map representation of an image. (a) A simple H-shape. (b) Distance map of (a)

the set make up an object centerline, it is not necessarily a one-pixel wide line. Thus, “thinness,” one of the required properties of a skeleton, is not met. Similar to previous methods, a priori knowledge of object boundaries is also required for topological thinning. In addition, different criteria for pixel removal produces different skeletons for the same object.

2.4 Ridge-Following

Ridge-following is a skeletonization method that works by following ridges on a topological surface. Ridges on topological surfaces correspond to object centerlines. Two examples of topological surfaces, used by ridge-following, are the distance map of a binary object and the intensity profile of a grayscale object. On an intensity profile, ridge points are the points with the highest intensity values. Similarly, on a distance map, ridge points have the highest distance values. The distance map of the H-shaped object in Figure 2.6(a) is illustrated in Figure 2.6(b), where the third dimension is the distance values of the pixels in the H-shaped object. The purpose of ridge-following methods is to extract the ridge pixels on topological surfaces.

An early method of ridge-following uses a technique known as deformable curves. Leymarie et al. introduced a skeletonization method using “Active Contour Models” or “Snakes” [8]. This method was developed to simulate Blum’s idea of grassfire propagation. Snakes are a deformable curve in the continuous space acting on a potential surface, such as an inverted distance map. Snakes are influenced by both external and internal forces. An example of an external force

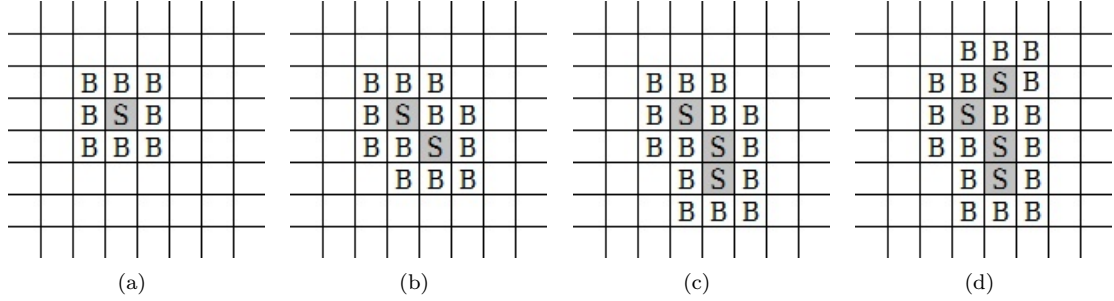


Figure 2.7. Ordered Region Growing. Four iterations of the ORG method are shown

is gravity acting on a snake to “pull” it down a potential surface. Internal forces affecting the snake are introduced by strings and rods which influence how the snake stretches and bends, respectively. A snake is initialized on an object boundary and is fixed to vertices. A snake’s goal is to reach a minimum energy state while being influenced by these internal and external forces. When a snake has reached an equilibrium, it stops moving. In effect, a snake “falls” towards the centerline of an object. There are some drawbacks to this method. The internal and external energies must be carefully defined to produce a good result. Additionally, objects with holes in them require additional snakes to be initialized on the boundaries of each hole.

Yim et al. proposed a ridge-following skeletonization method called Ordered Region Growing (ORG) [7]. The purpose of this method is to expose ridge points on a distance map that are contained within a well defined subset of pixels. The algorithm records ridge points in a result set S and searches for new ridge points in a well defined search-space. The algorithm begins by adding a seed point s , chosen inside an object, to the set S . A set of pixels B , the search-space, contains all the pixels bordering s as illustrated in Figure 2.7(a). From this set B , the highest intensity pixel b is added to S . The next step of the algorithm adds all neighbors of b to B , excluding neighbors already in S , effectively “growing” the search-space as illustrated in Figure 2.7(b). The final step then removes b from B . This process repeats until all pixels within the image have been processed. Figures 2.7(c) and (d) show two additional iterations of this algorithm. A weakness of this method is the need to carefully select an intensity threshold to avoid adding every pixel to S . Multiple seed points may also need to be chosen for different objects in an image. Finally, loops in the object, or objects that intersect (such as vessels that cross), cause disconnected skeletons.

Another method of ridge-following, proposed by Aylward et al., is called dynamic scale ridge traversal [13]. This method uses a basic ridge detection procedure to follow ridges along tubular objects and enhances the procedure with a “dynamic scale.” The dynamic scale adjusts the calculations used to detect ridge points. The scale has various advantages. It provides insensitivity to noise, helping to eliminate high-frequency variations in the data. It is also used to blur the object to help estimate its central ridge. This method continually adjusts the scale based on an estimation of the local radius of the object in order to improve ridge point detection accuracy. To begin the ridge-following procedure, this method randomly selects seed points located near

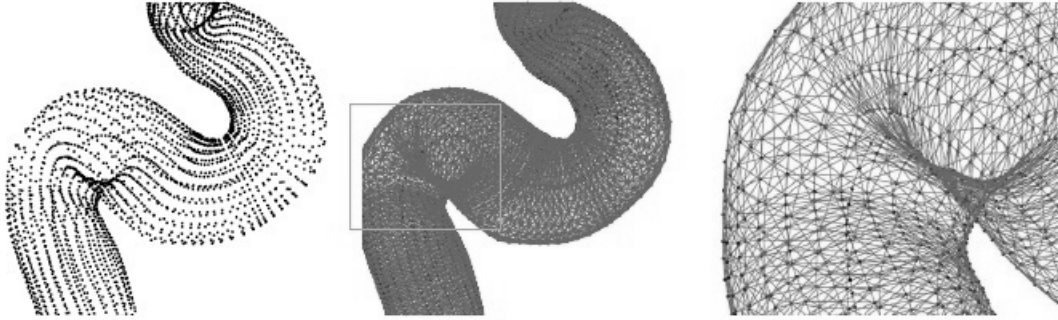


Figure 2.8. 3D representation of a vessel crawler

vessels. Random seed point selection, however, means that the results may be different each time an image is processed. Another drawback is that branching points were traversed correctly in only half of the tests where noise was present in the image, even though special heuristics were used for branching point traversal.

Ridge-following methods are not confined to two dimensional images. McIntosh et al. introduced a ridge-following method that operates in 3D discrete image space. It uses deformable organisms and artificial life decision-making processes to follow the centerlines of objects [14]. These “vessel crawlers,” illustrated in Figure 2.8, are made up of five modules: the cognitive center module, sensor module, behavioral module, locomotion module, and the geometry module. A vessel crawler’s organization into these five modules has advantages and disadvantages. It decouples various parameters allowing for interactive control of the vessel crawler in-process. However, the vessel crawlers are non-deterministic, potentially producing different results on subsequent processing of the same image. Additionally, this method is complicated to implement and requires extensive processing power.

Ridge-following methods, in general, share similar weaknesses. They require a priori knowledge of vessel boundaries either for initialization or for processing. Holes within objects can introduce erroneous results, or at least require special processing. They require user interaction such as selecting one or more seed points from which the ridge-following is initiated. If there are multiple objects within an image, multiple seed points may need to be selected to process the objects separately. These weaknesses prevent the automatic computation of skeletons.

Fast Skeletonization of Blood Vessels

The method is divided into four steps. In the first step, a matched filter is applied to the topological surface of an MRA image to enhance vessels and reduce image noise. In the second step, the algorithm extracts ridge points from vessels. In the third step, ridge points are connected to form skeletons. Furthermore, the third step ensures the skeletons adhere to the connectedness and thinness properties of skeletonization. In the final step, active contour models are applied to compute smooth skeletons in the continuous space.

The proposed skeletonization method has several advantages over existing methods. It requires no user intervention to select seed points, facilitating automation. It requires no a priori knowledge of vessel boundaries. It produces skeletons in a form that allows morphological parameters of vascular networks to be automatically measured. It processes images as a whole, skeletonizing multiple vascular networks simultaneously. Finally, if desired, skeletons can be smoothed in the continuous space to improve accuracy.

3.1 Enhancing Vessels

MRA images, as seen in Figure 1.1, contain noise that makes ridge extraction difficult. Therefore, the first step in the proposed method is to enhance the vascular networks and reduce noise, using matched filter [15].

Due to the MRA process, vascular networks appear darker than the surrounding area (see Figure 1.1). Moreover, a vessel cross section has an intensity distribution that is lower in the center and increases towards the boundaries. Though vessel cross sections differ greatly throughout an image, as shown in Figure 3.1, it is clear the cross section patterns are similar. Furthermore, they resemble a Gaussian curve. The goal, then, is to detect this pattern in an image and mark those pixels as part of a vessel. This is accomplished with the matched filter.

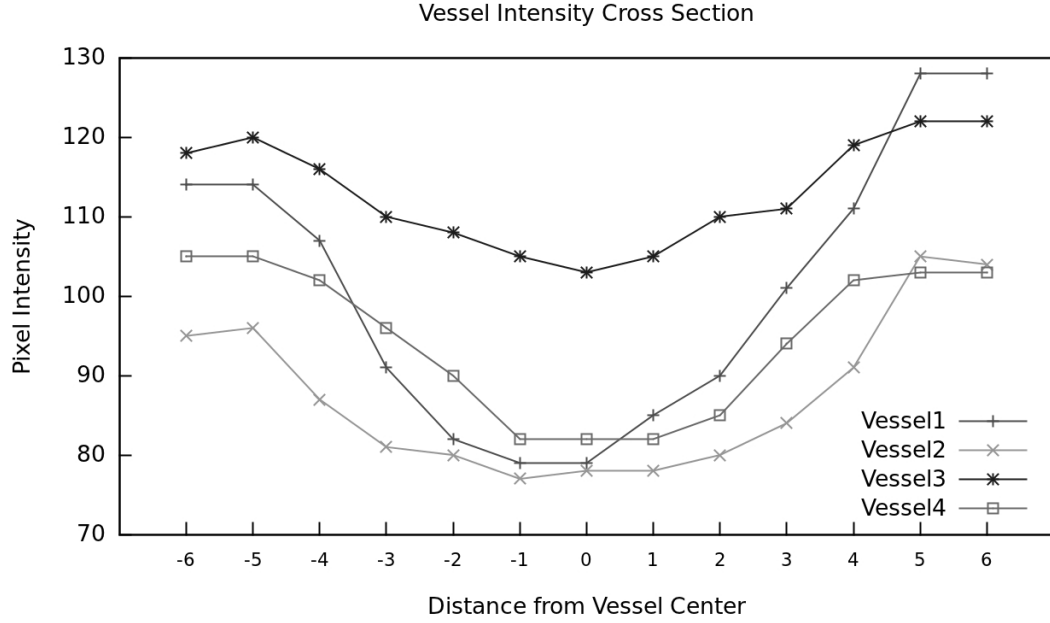


Figure 3.1. Intensity cross sections of vessels in an MRA image. Note the intensity distribution patterns are similar for all vessels

The matched filter is used to determine if a signal pattern matches an expected signal pattern, or optimal filter. The method convolves two functions, f and g , to produce a filter response:

$$(f \star g)(t) = \int_{-\infty}^{\infty} f(s) \times g(t - s) ds$$

where f is a signal pattern, g is the optimal filter, and \star is the convolution operator. When these two functions closely match, the output of their convolution produces a higher response.

Since the vessel cross sections resemble a Gaussian curve, a Gaussian distribution of intensity values is used as the optimal filter. The optimal filter is implemented as a kernel of 15×16 pixels as illustrated in Figure 3.2 [4, 15]. This kernel size produces the best results for the images used for experimentation. Notice that the intensity values of the kernel are lower towards the center and higher towards the edges, similar to the vessel cross sections described earlier.

In two dimensional image space, a vessel can be oriented in any direction. A kernel will produce a high filter response only when it is oriented in the same direction as the vessel. Thus, a set of twelve kernels, oriented in different directions, are used to detect vessels. An example of a kernel oriented at a 45° angle is illustrated in Figure 3.2(b) [15]. Additionally, notice that the kernels are “short.” Though vessels are tortuous, it is assumed they have a fixed direction and width for a short length [4]. Short kernels allow detection of vessels on curves and branching points.

The matched filter method convolves every image pixel with each kernel, producing multiple filter responses per pixel. The highest filter response is accepted as the overall filter response for

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	4	3	2	1	-2	-5	-6	-5	-2	1	2	3	4	0	0
0	4	3	2	1	-2	-5	-6	-5	-2	1	2	3	4	0	0
0	4	3	2	1	-2	-5	-6	-5	-2	1	2	3	4	0	0
0	4	3	2	1	-2	-5	-6	-5	-2	1	2	3	4	0	0
0	4	3	2	1	-2	-5	-6	-5	-2	1	2	3	4	0	0
0	4	3	2	1	-2	-5	-6	-5	-2	1	2	3	4	0	0
0	4	3	2	1	-2	-5	-6	-5	-2	1	2	3	4	0	0
0	4	3	2	1	-2	-5	-6	-5	-2	1	2	3	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(a)

0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0
0	0	0	0	0	4	4	3	0	0	0	0	0	0	0	0
0	0	0	0	4	4	3	2	0	0	0	0	0	0	0	0
0	0	0	4	4	3	2	0	-2	-4	0	0	0	0	0	0
0	0	4	4	3	2	0	-2	-4	-5	-6	0	0	0	0	0
0	4	4	3	2	0	-2	-4	-5	-6	-5	-4	0	0	0	0
4	4	3	2	0	-2	-4	-5	-6	-5	-4	-2	0	0	0	0
0	3	2	0	-2	-4	-5	-6	-5	-4	-2	0	2	3	0	0
0	0	0	-2	-4	-5	-6	-5	-4	-2	0	2	3	4	4	0
0	0	0	-4	-5	-6	-5	-4	-2	0	2	3	4	4	0	0
0	0	0	0	-6	-5	-4	-2	0	2	3	4	4	0	0	0
0	0	0	0	0	-4	-2	0	2	3	4	4	0	0	0	0
0	0	0	0	0	0	0	2	3	4	4	0	0	0	0	0
0	0	0	0	0	0	0	0	3	4	4	0	0	0	0	0
0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0

(b)

Figure 3.2. Matched filter kernels. (a) A kernel resembling a Gaussian distribution. (b) The kernel in (a) oriented at a 45° angle

the pixel [4]. Figure 3.3(a) shows an original MRA image while (b) shows the result of applying matched filter to (a). Notice that the vessels have been enhanced and noise has been reduced. Additionally, a vessel's intensity is high at the centerline and decreases towards the boundaries. This is due to a pixel's filter response being used as the new pixel intensity in the matched filter image.

3.2 Extracting Ridge Points

Due to the matched filter process, the intensity distribution of vessel cross sections are now bright in the center and diminish towards the vessel boundaries. The next step is to extract ridge points on the topological surface using the scanline method. We use the matched filter responses from the previous section as the topological surface. Note that the topological surface can be the distance maps for binary images or the intensity profile surfaces for grayscale images. The distance map of the binary object in Figure 3.4(a) is illustrated in (b). The third dimension

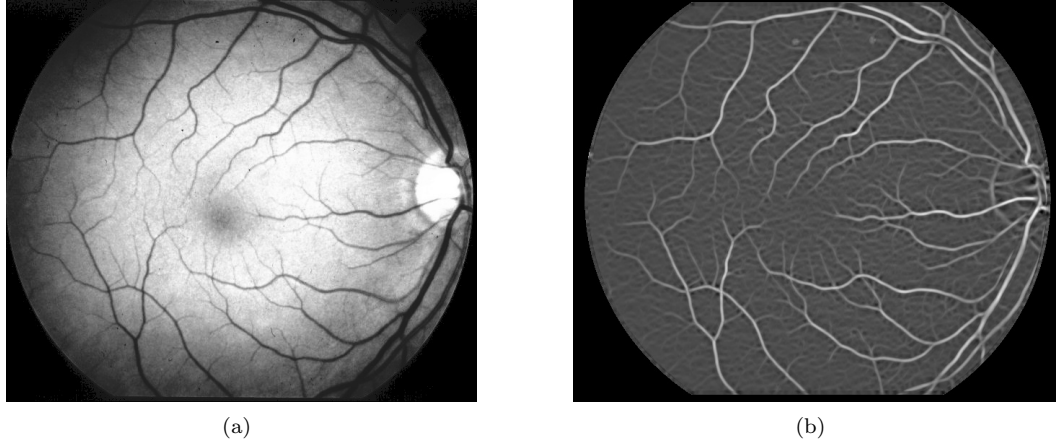


Figure 3.3. A matched filter image. (a) An MRA image of a retina. (b) The result of processing (a) with the matched filter

represents the distance of an object pixel from the nearest object boundary. An example of an intensity profile surface is illustrated in Figure 3.5. Figure 3.5(b) is the intensity profile surface of (a).

The main idea behind the extraction of the ridge points is based on two observations: ridge points are the points on the ridges of a topological surface and a ridge is where a discontinuity of the gradient of the surface occurs. To be a ridge point, the point must be a local maximum on a line passing through that point on the topological surface.

Consider a line, called a scanline, through a point on a topological surface. If the point is a local maximum, the surface values of its two opposite neighboring points are less than the given point's surface value and the directions of the two opposite neighbors' gradient vectors projected onto the scanline must be opposite, pointing to the given point. Further, if we choose the scanlines parallel to the x - and y -axis, the projection of the gradient vector onto the scanline becomes trivial:

$$M_x : (v_x, v_y) \mapsto (\text{sign}(v_x), 0)$$

$$M_y : (v_x, v_y) \mapsto (0, \text{sign}(v_y))$$

where v_x and v_y are x and y components of a given vector, $\text{sign}()$ is a function that returns the sign of its argument, and M_x and M_y are the mappings onto the scanlines parallel to the x - and y -axes, respectively. Figure 3.4(c) and (d) shows the results of the mapping applied to the topological surface in (b) of a simple H shaped object. Note that a scanline cannot detect a ridge parallel to itself, e.g., the ridge of the horizontal bar in the middle of the H shape is not detected in Figure 3.4(c). Similarly, the ridges on the two vertical bars of the H shape are not captured in Figure 3.4(d). However, the two scanlines combined together reveal the existence of all the ridges not only in the directions of the scanlines but also in arbitrary directions. These facts indicate that at least two scanlines are needed to detect any arbitrarily oriented ridges. In fact, only two

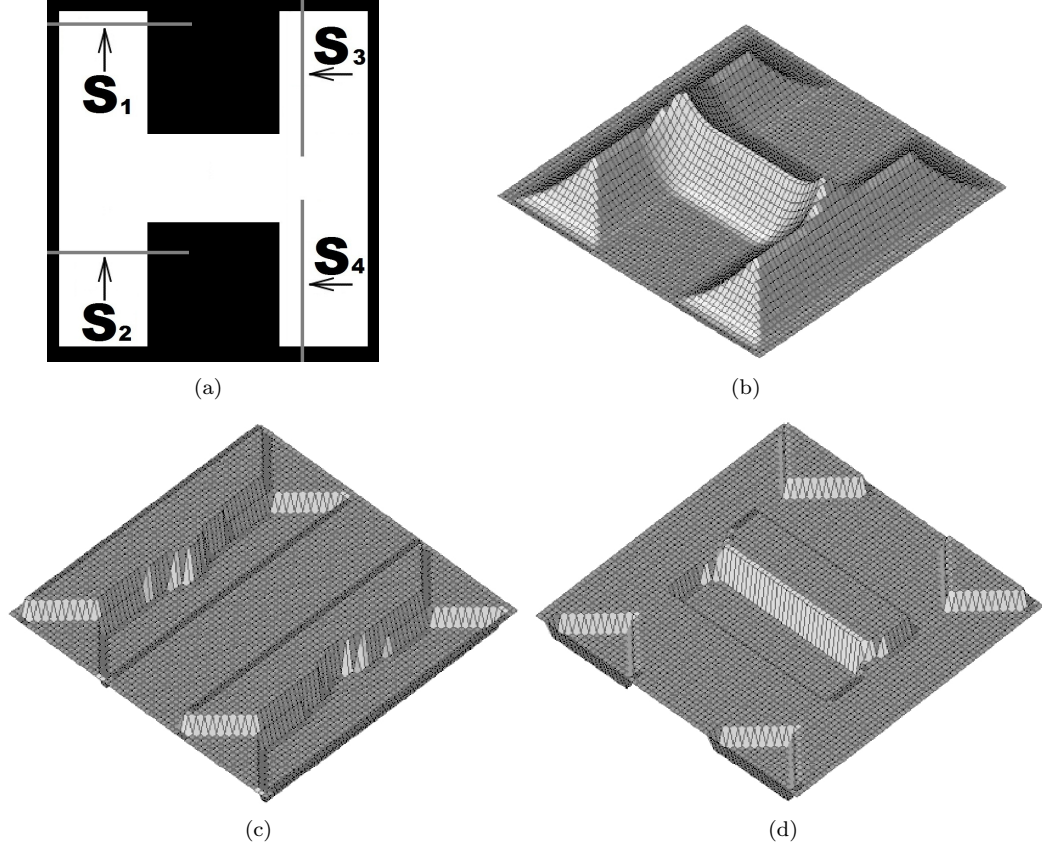


Figure 3.4. Distance map. (a) Simple H-shaped object. The labels, arrows, and gray lines are not part of the object. (b) Distance surface of (a). (c) Gradient vectors are projected onto scanline along x -axis. (d) Gradient vectors are projected onto scanline along y -axis

orthogonal scanlines should suffice to detect all the ridges with arbitrary orientations. This is because a scanline can detect any ridge that crosses the scanline. In other words, a scanline only fails to detect a ridge that is parallel to itself. However, such a ridge must cross another scanline whose orientation is sufficiently different from the first scanline. Therefore, only two scanlines with sufficiently different orientations will detect all the ridges present on the topological surface. With this observation, the problem of detecting the ridge points is reduced to the problem of detecting the sign changes of the gradient vectors projected onto two scanlines with sufficiently different orientations. This suggests a simple scanline algorithm using the horizontal and vertical scanlines, S_x and S_y , respectively.

Before describing the algorithm, we need to define the prominent sign changes. A prominent sign change is the change in the signs of projected gradient vectors on a scanline that indicates the existence of a ridge. Figure 3.6(a) lists all the possible cases of the sign changes between two neighboring points on a scanline. Among the six cases, $+-$ shows a strong indication of the existence of a ridge between the two points. On the other hand, $-+$ shows little or no indication. This is because $-+$ can only occur at the section where two tapering sub shapes (e.g., in terms

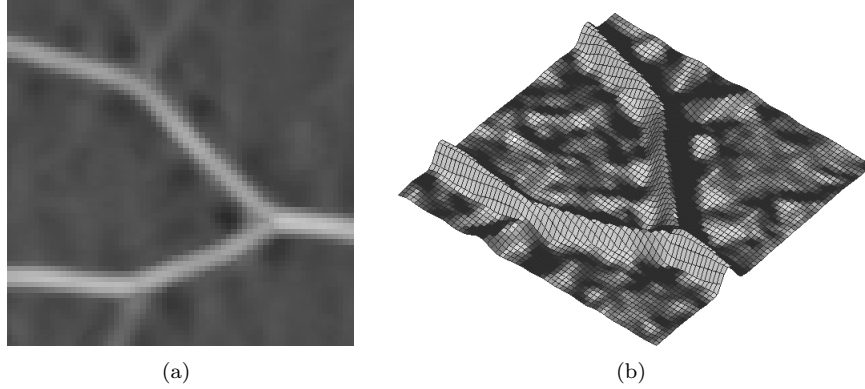


Figure 3.5. Intensity profile surface. (a) Vessel segment and bifurcation from grayscale image. (b) Intensity profile surface of (a)

				+-	+o	-+	-o	o+	o-
				(a)					
+[o···]+		-[o···]+		o[+···]-		o[-···]+			
+[o···]-		-[o···]-		o[+···]o		o[-···]o			
(b)		(c)		(d)		(e)			

Figure 3.6. Sign change patterns of the projected gradient vectors on a scanline

of elementary shape descriptors defined by Blum, such shapes as wedges, flares, or cups [6]) meet at their closing ends forming a concave boundary in such a way that the ridge of the shape is nearly parallel to the given scanline. Such a ridge, however, will appear as $+-$ on the other orthogonal scanline, allowing us to ignore the case $-+$.

In order to determine whether they indicate the existence of ridges, the remaining four patterns (i.e., $+o$, $-o$, $o+$, $o-$) need to be expanded to more than two neighboring points. These are illustrated in Figure 3.6(b), (c), (d), and (e), where $[s···]$ indicates one or more occurrences of the sign s . It is easy to see that the patterns in Figure 3.6(b) are the indications of the ridge existence. For example, $+[o···]-$ occurs when S_x , the horizontal scanline, is at the position of S_1 in Figure 3.4(a) and the two ridges can be found in Figure 3.4(b) at the points of $+$ and $-$. Thus, the points corresponding to $+$ and $-$ are marked as the ridge points. Note that the pattern $+o-$ should be handled as a special case of $+[o···]-$ that strongly indicates the existence of a ridge as does $+-$. The $+o-$ and $-+$ patterns can be found when S_x is located at the position of S_2 in Figure 3.4(a). A ridge exists at the point corresponding to o in the case of $+o-$ and only that point is marked as a ridge point. In the case of $+-$, however, a ridge point is at a non-integer coordinate between the two points of $+$ and $-$, with one of the two points having a larger value being marked as a ridge point since we are searching for local maxima on the topological surface. Now, consider S_y , the vertical scanline, at the position of S_3 in Figure 3.4(a). At that position, a $+[o···]+$ is found. The algorithm then marks the point of the first $+$ as a ridge point and

continues the search of the patterns starting from the point of the last \circ since $\circ+$ is handled as a special case as will be discussed below.

Now, consider the cases in Figure 3.6(c). $-\circ\cdots-$ can be found when S_y is located at the position of S_4 in Figure 3.4(a). The point of the last $-$ is marked as a ridge point. Notice that any scanline must start with the pattern $\circ\cdots+$ indicating the leftmost or topmost edge and must end with the pattern $-\circ\cdots$ indicating the rightmost or bottommost edge. Therefore, in order for $-\circ\cdots-$ to occur, there must be a $+$ followed by zero or more \circ 's in front of it, which is one of the patterns in Figure 3.6(b). Hence, the first $-$ in $-\circ\cdots-$ must have been processed by other cases and is ignored here. The next pattern $-\circ\cdots+$ suggests a valley or basin and, for the same reason as the pattern $-+$, shows little or no indication of ridge existence. Thus, it is not considered a prominent sign change.

Finally, the patterns in Figure 3.6(d) and (e) are determined by other patterns previously discussed. For example, consider $\circ[+\cdots]-$. If this is the first sign change detected on a scanline indicating the leftmost or topmost edge, then it is equivalent to $+ -$. Otherwise, there must be a $+$ or $-$ followed by zero or more \circ 's in front of it, and should be processed by $+\circ\cdots+$ or $-\circ\cdots+$ followed by $+ -$. The same arguments can be applied to all the remaining cases. The result is that only the following 5 patterns need be considered as the prominent sign changes: $+ -$, $+\circ-$, $+\circ\cdots-$, $+\circ\cdots+$, and $-\circ\cdots-$.

With the 5 prominent sign changes identified, the proposed algorithm scans an image from top to bottom with S_x and from left to right with S_y , looking for the prominent sign changes. For each scan, it searches for the prominent sign changes from left to right for S_x and from top to bottom for S_y . When it detects one, the algorithm marks the point as a ridge point, and continues the scan. After the scanning process is done in both the x and y directions, the algorithm simply returns the set of the marked points for further processing. Figure 3.7 illustrates the scanline method applied to Figure 3.3(b).

At this point, we have a set of ridge points detected by the scanline algorithm. However, because the scanline algorithm is greedy, it returns both true-positive and false-positive ridge points. False-positive ridge points are a result of noise within the image. The next step is to eliminate false-positives from the candidate ridge points by applying the ratio filter to the candidate ridge points. This filter compares the intensity of a detected ridge point to those of the nearest opposing boundary points. In general, the ratio between the intensity of a ridge point and those of its boundary points is higher for vessels than for local noise. Thus, this step looks for the intensity ratio of the ridge point to boundary point above a threshold. Those ridge points, whose intensity ratio exceeds the threshold, are considered true-positive ridge points. Otherwise, the ridge point is eliminated as a false-positive due to noise.

Let I_C , I_L , and I_R be the intensity values of a ridge point and its left and right boundary points, respectively. The ratio values R_l and R_r , then, are defined as:

$$R_l = I_C / I_L$$

$$R_r = I_C / I_R$$

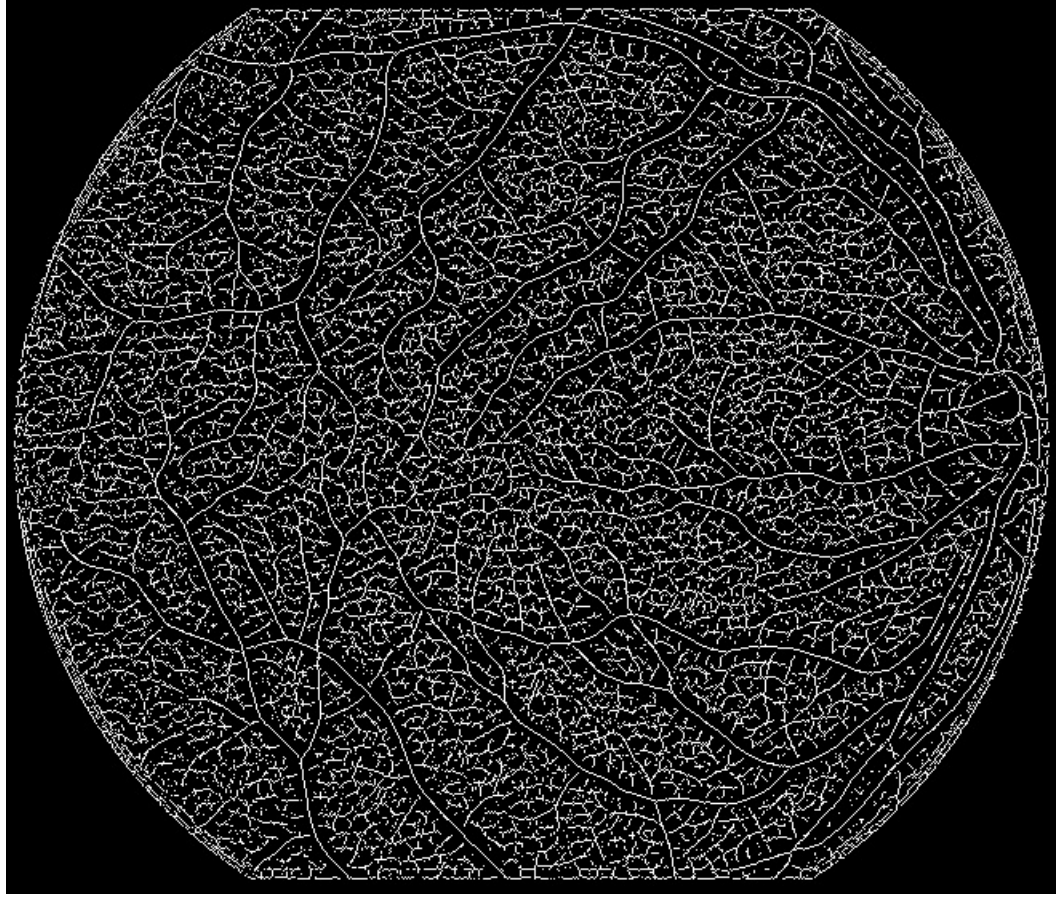


Figure 3.7. The result of applying the Scanline algorithm to Figure 3.3(b)

To determine whether the point R is a true-positive ridge point, one or both of the ratio values must exceed a pre-specified threshold T . The threshold is selected by examining the images used during experimentation.

$$f(T, R_l, R_r) = \begin{cases} true, & \text{if } \max(R_l, R_r) \geq T \\ false, & \text{otherwise.} \end{cases}$$

To better illustrate the ratio filter, Figure 3.8 shows a cross section of a vessel ridge and a noise ridge from a matched filter image. On the left is a ridge typically caused by noise (I_C^2); on the right is a ridge caused by a vessel (I_C^1). The intensity ratios I_C/I_R and I_C/I_L illustrate the difference between noise and vessels in the ratio filter's context.

The ratio filter examines candidate ridge points detected by the scanline algorithm, eliminating false-positives caused by noise. An example of the ratio filter method applied to Figure 3.7 is illustrated in Figure 3.9. Note, in the figure, that most of the noise have been eliminated while the true ridge points are intact.

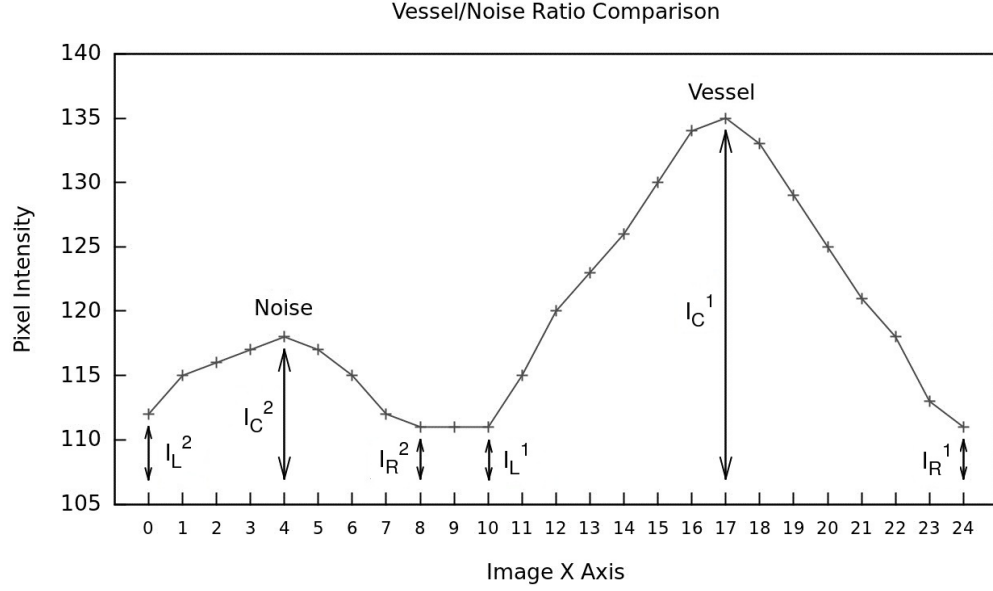


Figure 3.8. Intensity cross section showing a noise ridge (left) and a vessel ridge (right) where I_C is the intensity of the ridge pixel, and I_L , and I_R are the left and right boundary pixel intensities, respectively. The intensity ratios I_C/I_L and I_C/I_R for the vessel are greater than the intensity ratios for the noise.

3.3 Building Skeletons

The scanline and ratio filter algorithms generate an unstructured set of ridge points. The goal of this section is to explain how to put the ridge points into a graph representation that will facilitate further processing and automated measurement of morphological parameters. We present a simple 2-pass linking algorithm that constructs the graph representation of the ridges in the discrete image space by tracing the maximum gradient paths on the topographical surface of a given image.

The scanline algorithm may generate gaps between the ridge points detected due mainly to its discrete nature as well as its failure at planar points. For example, when it detects a prominent sign barrier on a scanline, the algorithm selects the point with the maximum surface value among the involved points. This selection process may pose the problem of the discontinuous ridge points, i.e., gaps. Due to the gaps, the linking algorithm consists of two passes. In the first pass, the ridge points are linked together. The second pass detects and repairs the gaps caused by the scanline algorithm.

Let T be the set of candidate ridge points returned by the scanline method and S be the set of ridge points returned by the ratio filter method. That is:

$$S = \{p \in T \mid R_p \geq t\}$$

where R_p refers to the ratio value of a point p and t is the user specified ratio threshold value.

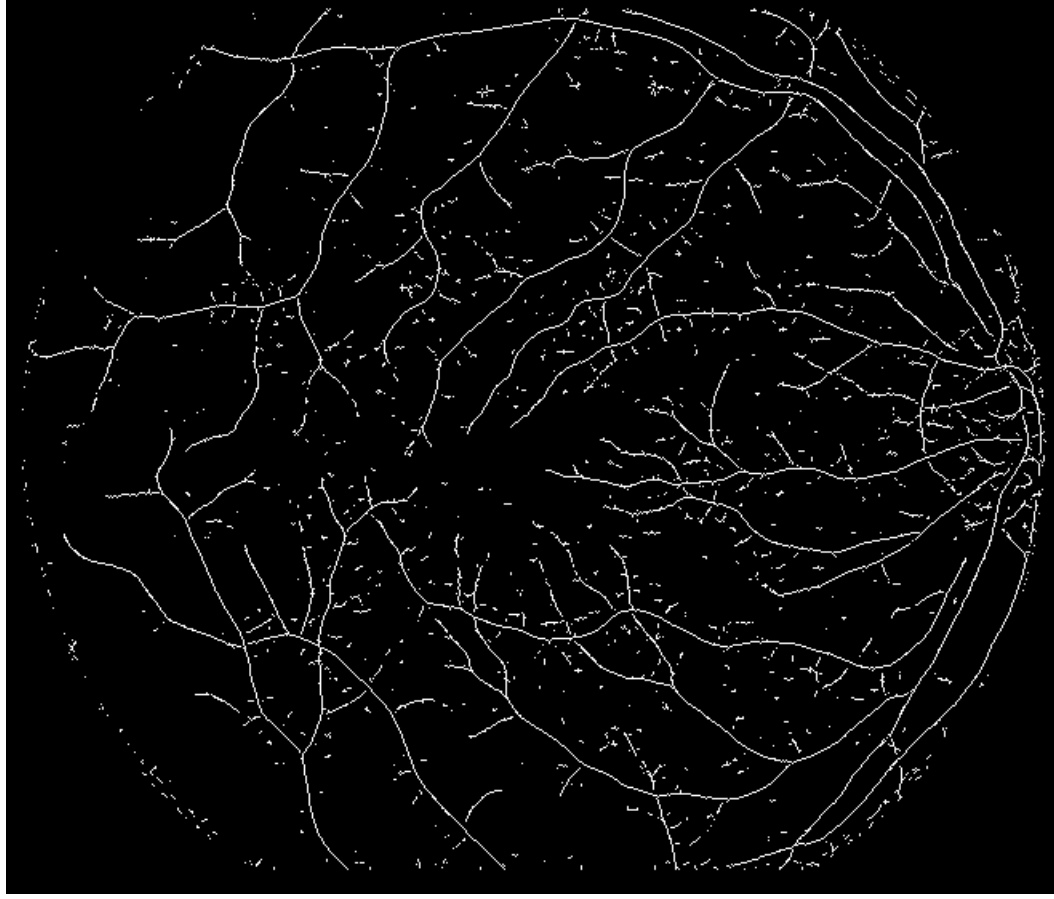


Figure 3.9. Ratio filter applied to Figure 3.7

Also, let L be the set of the links, where a link is represented as a pair of points $\langle p, q \rangle$. The linking algorithm attempts to connect a ridge point, p , to its neighboring point, q , using one of the three types of links:

$$l_1(p) = \langle p, q \rangle, \quad q = \max(\text{neighbors}(p) \cap S)$$

$$l_2(p) = \langle p, q \rangle, \quad q = \max(\text{augment}(p) \cap S)$$

$$l_3(p) = \langle p, q \rangle, \quad q = \max(\text{augment}(p))$$

where,

$\text{neighbors}(p)$: 8-neighbors of p ,

$\max()$: the point with the maximum surface value among its arguments,

$\text{augment}(p)$: the set of the candidate augmentation points at p , i.e.,

$$\text{augment}(p) = \text{neighbors}(p) / \text{neighbors}(p')$$

where, p' is the point previously linked to p .

If two or more points share the maximum surface value, $\max()$ selects the point with the maximum surface value for l_1 or the point that forms an angle close to 180 degrees to the link $\langle p', p \rangle$ (i.e. $\angle p'pq$) for others, prohibiting the sharp turns. Note that using $\text{augment}(p)$ in l_2 and l_3 also prevents the links from making the sharp turns in their directions.

In the first pass, each $p_n \in S$ is linked to at most two neighboring points. The first link is made applying l_1 and the second l_2 . Then, L is updated recursively as:

$$\begin{aligned} L_0 &= \emptyset \\ L_n &= L_{n-1} \cup \{l_1(p_n)\} \cup \{l_2(p_n)\}, \quad n = 1, \dots, |S| \end{aligned}$$

Furthermore, the first pass promotes skeleton thinness. A thin line is one that has a single pixel width, meaning a pixel in the line should have no more than two neighbors.

Due to the gaps, the first pass does not always produce a completely connected graph for a connected object. Instead, it converts the set of the ridge points into the set of the ridge lines. The second pass tries to link these disconnected ridge lines, filling the gaps by tracing the maximum gradient paths on Q , the set of all points on the topographic surface. Note that the tracing in one gapped area normally completes in a few steps since the gap usually occurs in a small region. Let EP be the set of the end-points of the ridge lines, i.e.,

$$EP = \{p \in S \mid \text{degree}(p) = 1\}$$

where, $\text{degree}(p)$ is the number of the links sharing p . In the second pass the algorithm attempts to apply l_2 then l_3 to $p_m \in EP$. If l_3 is used to make the link, it repeats the process from the newly selected point until l_2 is applied, in which case the trace is accepted, or until the trace falls into the background region, in which case the trace is ignored. Formally, let $p_{m,r}$ and $C_{m,r}$ be the point chosen and the candidate list of the links traced so far at the r -th iteration of the tracing originated from $p_{m,0} = p_m \in EP$, i.e.,

$$\begin{aligned} C_{m,0} &= \emptyset \\ C_{m,r} &= C_{m,r-1} \cup \{l_2(p_{m,r}) \text{ or } l_3(p_{m,r})\} \end{aligned}$$

When $C_{m,r}$ is accepted, L and S are updated such that

$$\begin{aligned} L_m &= L_{m-1} \cup C_{m,r}, \quad L_0 = L_n^1 \\ S_m &= S_{m-1} \cup \text{flat}(C_{m,r}), \quad S_0 = S^1 \end{aligned}$$

where L_n^1 and S^1 refer to the resulting L and S from the first pass and $\text{flat}()$ the set of points used in the links in its argument. After the second pass, we have the graph representation of the ridges (i.e., L_m), in which each disconnected object on Q is represented by a separate subgraph. From the resulting L_m and S_m , it is easy to obtain the various properties of the graph. For example, the sets of the branching points (BP), the normal points (NP), and the branch end

points (EP) can be defined as:

$$BP = \{p \in S \mid \text{degree}(p) \geq 3\}$$

$$NP = \{p \in S \mid \text{degree}(p) = 2\}$$

$$EP = \{p \in S \mid \text{degree}(p) = 1\}$$

With the definition of the path between two points as:

$$\text{path}(p_1, p_n) = \{\langle p_1, p_2, \dots, p_n \rangle \mid \langle p_i, p_{i+1} \rangle \in L, 1 \leq i \leq n-1\},$$

we can define the branches of the graph as:

$$B = \{b = \text{path}(p, q) \mid p, q \in (BP \cup EP), i \in \text{path}(p, q) / \{p, q\}, i \in NP\}.$$

Two branches, b_1 , and b_2 , are connected if there is a path between their end points and are denoted by $\text{conn}(b_1, b_2)$. The subgraph containing a branch, b , can then be defined as:

$$G_s(b) = \{g = \cup i \mid \text{conn}(b, i), i \in B\}.$$

Topological surfaces are sensitive to noise. A small noise on the object boundary can initiate a spurious skeleton branch and a noise inside an object may alter the topological surface dramatically. Furthermore, noise outside of an object creates false-positive branches that may not be removed by the ratio filter described in the previous section. These false-positive branches also tend to be short. Therefore, the final step in this section is a pruning operation. The pruning algorithm measures the length of each branch, ignoring branching points and crossover points. If a branch is sufficiently short, it is removed completely. The maximum pruning length is dependent on the size of the image and is generally less than ten pixels for the images used during experimentation. In Figure 3.10(a), many false-positive branches are visible. In Figure 3.10(b), the false-positive branches are eliminated after applying the pruning algorithm.

3.4 Smoothing Skeletons

The skeleton graph computed in the previous section is a discrete approximation to the skeleton in the continuous space. When a smooth graph in the continuous space is desirable, the skeleton graph is regarded as a collection of deformable splines and the active contour model is applied to it [16, 17].

Let $v(s, t) = (x(s, t), y(s, t))$ represent a deformable curve with $s \in \Omega = [0, 1]$ and $t \in T$, where s and t are the spatial and time parameters, respectively. The potential energy functional $U(v)$ of the curve is defined as

$$U(v) = \frac{1}{2} \int_{\Omega} [S(v) + P(v)] ds \quad (3.1)$$

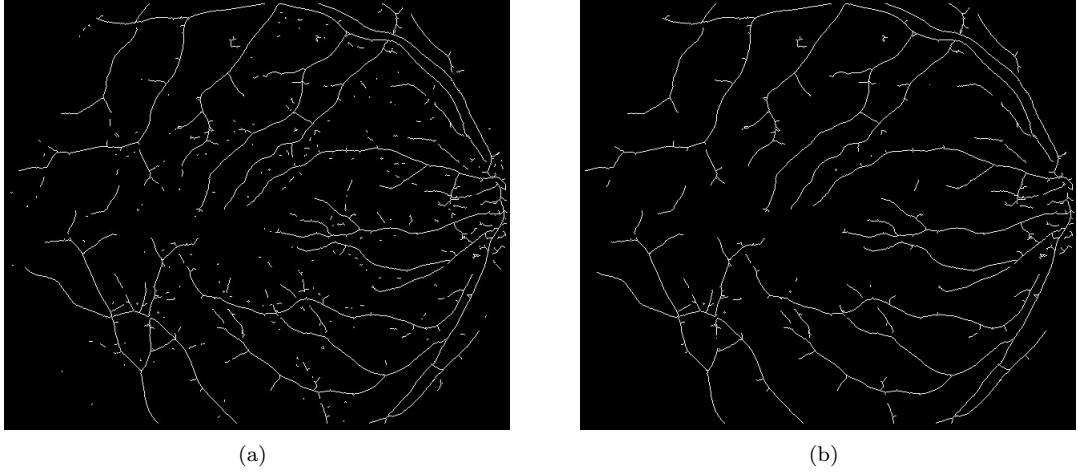


Figure 3.10. Pruning spurious branches. (a) An unpruned skeleton with many false-positive branches. (b) The skeleton after pruning. Note the false-positive branches are eliminated

where $S(v)$ is the internal deformation energy, which allows the snakes to resist stretching and bending, and $P(v)$ is the external energy potentials coming from a given image, simulating gravity that draws a snake down a potential surface, and other constraints such as spring-like forces. The internal deformation energy is defined in [16] as

$$S(v) = \omega_1(s)|v_s|^2 + \omega_2(s)|v_{ss}|^2 \quad (3.2)$$

where, $v_s = \partial v / \partial s$ and $v_{ss} = \partial^2 v / \partial s^2$. The first-order term makes the snake resistant to stretching, while the second-order term resists bending. With the kinetic energy functional $T(v)$ defined as

$$T(v) = \frac{1}{2} \int_{\Omega} \mu |v_t|^2 ds \quad (3.3)$$

the Lagrangian L becomes

$$L = T - U = \int_{\Omega} \frac{1}{2} [\mu |v_t|^2 - \omega_1(s)|v_s|^2 - \omega_2(s)|v_{ss}|^2 - P(v)] ds \quad (3.4)$$

where, μ is the mass density and $v_t = \partial v / \partial t$. The motion of the deformable curve during the time interval $[t_0, t_1]$ under the influence of nonconservative forces is described by the function $v(s, t)$ for which the functional

$$I = \int_{t_0}^{t_1} (T - U + W_{nc}) dt = \int_{t_0}^{t_1} L dt + \int_{t_0}^{t_1} W_{nc} dt \quad (3.5)$$

is the minimum, where W_{nc} is the work done by nonconservative forces. Note that we consider the viscous damping as the only nonconservative force acting on the system. The viscous damping

force can be conveniently handled by the use of Rayleigh's dissipation function defined as

$$\mathcal{F} = \frac{1}{2} \int_{\Omega} \gamma |v_t|^2 ds \quad (3.6)$$

where, γ is the viscosity factor [18]. The application of the minimum energy principles requires the stationary condition for the functional I . In such a case, the Euler-Lagrange equation of motion for the nonconservative system becomes the following (see [18] for details):

$$\mu x_{tt} + \gamma x_t - \frac{\partial}{\partial s}(\omega_1(s)x_s) + \frac{\partial^2}{\partial s^2}(\omega_2(s)x_{ss}) = -\frac{1}{2}P_x(v) \quad (3.7)$$

$$\mu y_{tt} + \gamma y_t - \frac{\partial}{\partial s}(\omega_1(s)y_s) + \frac{\partial^2}{\partial s^2}(\omega_2(s)y_{ss}) = -\frac{1}{2}P_y(v) \quad (3.8)$$

The discretization of Equations (3.7) and (3.8) can be performed by defining the dynamic deformable model as a physical system of n nodes connected in series and represented with the generalized coordinate vector $\mathbf{q} = [q_1, q_2, \dots, q_n]^T$. The generalized form of Euler-Lagrange equation of motion under the influence of the viscous damping force as the only nonconservative force acting on the system is given in [18] as

$$\frac{\partial}{\partial t} \left(\frac{\partial L}{\partial \dot{q}_k} \right) - \frac{\partial L}{\partial q_k} + \frac{\partial \mathcal{F}}{\partial \dot{q}_k} = 0, \quad k = 1, 2, \dots, n \quad (3.9)$$

The discrete version of T , U , and \mathcal{F} can be written in quadratic matrix form as

$$\begin{aligned} T &= \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} \\ U &= \frac{1}{2} (\mathbf{q}^T \mathbf{K} \mathbf{q} + P(\mathbf{q})) \\ \mathcal{F} &= \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{D} \dot{\mathbf{q}} \end{aligned}$$

where \mathbf{M} , \mathbf{D} , and \mathbf{K} are mass, damping, and stiffness matrices respectively. It follows that

$$L = T - U = \frac{1}{2} (\dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} - \mathbf{q}^T \mathbf{K} \mathbf{q} - P(\mathbf{q}))$$

Therefore, substituting

$$\frac{\partial}{\partial t} \left(\frac{\partial L}{\partial \dot{\mathbf{q}}} \right) = \mathbf{M} \dot{\mathbf{q}}, \quad \frac{\partial L}{\partial \mathbf{q}} = -\mathbf{K} \mathbf{q} - \nabla P, \quad \text{and} \quad \frac{\partial \mathcal{F}}{\partial \dot{\mathbf{q}}} = \mathbf{D} \dot{\mathbf{q}}$$

into Equation (3.9), we obtain the linearized equation of motion in matrix form as

$$\mathbf{M} \ddot{\mathbf{q}} + \mathbf{D} \dot{\mathbf{q}} + \mathbf{K} \mathbf{q} = -\nabla P \quad (3.10)$$

The matrices \mathbf{M} , \mathbf{D} , and \mathbf{K} are sparse and banded diagonal matrices due to the discretization of the continuous model using finite differences [6]. Equation (3.10) can be transformed into a

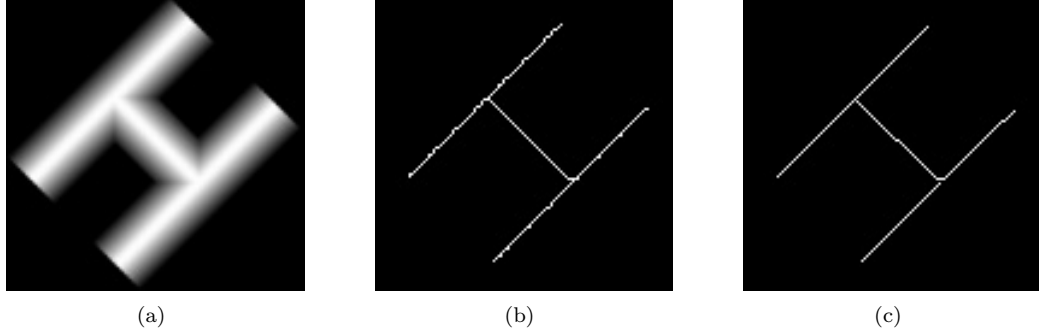


Figure 3.11. Results of the smoothing method. (a) A simple H-shaped object. (b) The initial skeleton obtained by applying the extract and graph building methods to (a). (c) Deformed skeleton of (b) after 30 iterations of smoothing steps

system of the first-order differential equations by denoting the nodal velocities $\dot{\mathbf{q}}$ with \mathbf{v} as

$$\begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} -\mathbf{M}^{-1}\mathbf{D} & -\mathbf{M}^{-1}\mathbf{K} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{q} \end{bmatrix} + \begin{bmatrix} -\mathbf{M}^{-1}\nabla P \\ \mathbf{0} \end{bmatrix} \quad (3.11)$$

Depending on the design of the energy functional, the deformable models can be used to seek various image features. In [8], a dynamic snake is used on inverted topological surfaces, where the snake falls down the surface to reduce its energy, seeking the valleys of the inverted surfaces. To accomplish this, a potential field energy

$$E_{\text{field}}(\mathbf{v}(s, t)) = \mu \mathcal{G} z(\mathbf{v}(s, t)) \quad (3.12)$$

is added to the external energy $P(v)$ in Equation (3.1), where μ is the constant mass density of the snake, \mathcal{G} the magnitude of the gravitational acceleration, and $z(\mathbf{v}(s, t))$ the height (or potential value) of the topological surface at a snake point at time t . In order to compute the smooth skeleton graphs following this approach, the coarse graph is regarded as a collection of dynamic snakes and deformed on the inverted topological surface. For each branch in the skeleton graph, a snake is initialized on the inverted topological surface. Then, both ends of the snake are anchored to the corresponding branching or branch end points with springs. A spring between two points, p_0 , and p_1 , can be created by simply adding $-k(p_0 - p_1)^2$ to the external energy $P(v)$ in Equation (3.1). Note that all snakes initialized are open snakes. The open snake was implemented by introducing a positional discontinuity between the last and first points of the snake [19].

Figure 3.11 illustrates the results of the smoothing method applied to a simple H-shaped object skeleton. The original H-shaped image is illustrated in (a). Figure 3.11(b) shows the skeleton before smoothing is applied. As expected, the initial graph is very coarse since it is constructed in the discrete image space. The result after 30 iterations of the smoothing steps is shown in (c), where we can see that the coarse edges are smoothed.

In the next chapter we present the results of applying all of the proposed method's steps to MRA images of human retinas. In addition, we compare the proposed method's results to the results of other methods.

Results

To evaluate the proposed method, we used the publicly available STARE (Structured Analysis of the Retina) database of retinal images [4]. The images were captured by a TopCon TRV-50 fundus camera at 35° field of view and digitized to generate 700×605 pixel images with 24 bits per pixel (RGB). To provide a ground-truth segmentation of vessels, an expert observer hand-labeled the vessels, which may take several hours depending on the image [4]. A majority of pixels belonging to a vessel can be easily labeled by a human observer. However, pixels on vessel boundaries, in very small vessels, and near pathology can be difficult to accurately identify. Therefore, a second expert observer hand-labeled the vessels to provide additional ground-truth segmentations for comparison.

We applied the proposed method to the images in the STARE database and a sample of the results are illustrated in Figure 4.1. The figure shows the original images on the left and the results of the proposed method applied to those images on the right. From the figure, we can see that all the salient features have been successfully detected and skeletonized, along with many small vessels that are not visually obvious. Notice also that the skeletons are thin. Furthermore, these skeletons were smoothed with the active contour models discussed in the previous chapter. However, due to the high resolution of the images, the visual difference of the skeletons when smoothed is imperceptible.

The experiments were performed on a common personal computer with a 2.67GHz Intel Core i7 processor with 12GB of memory. Table 4.1 shows the average processing time, in milliseconds, of each step of the proposed method applied to the images from the STARE database. In the *Enhancing* column of the table, the matched filter method takes the majority of the processing time at roughly 1300 milliseconds on average. This is due to the fact that each pixel in an image must be processed by twelve kernels. In the *Extracting* column, which includes the scanline and ratio filter methods, processing takes about 200 milliseconds on average. In the *Building* column, which includes graph building and pruning, processing takes about 150 milliseconds. In the *Smoothing* column, which consists of the active contour method, processing takes less than

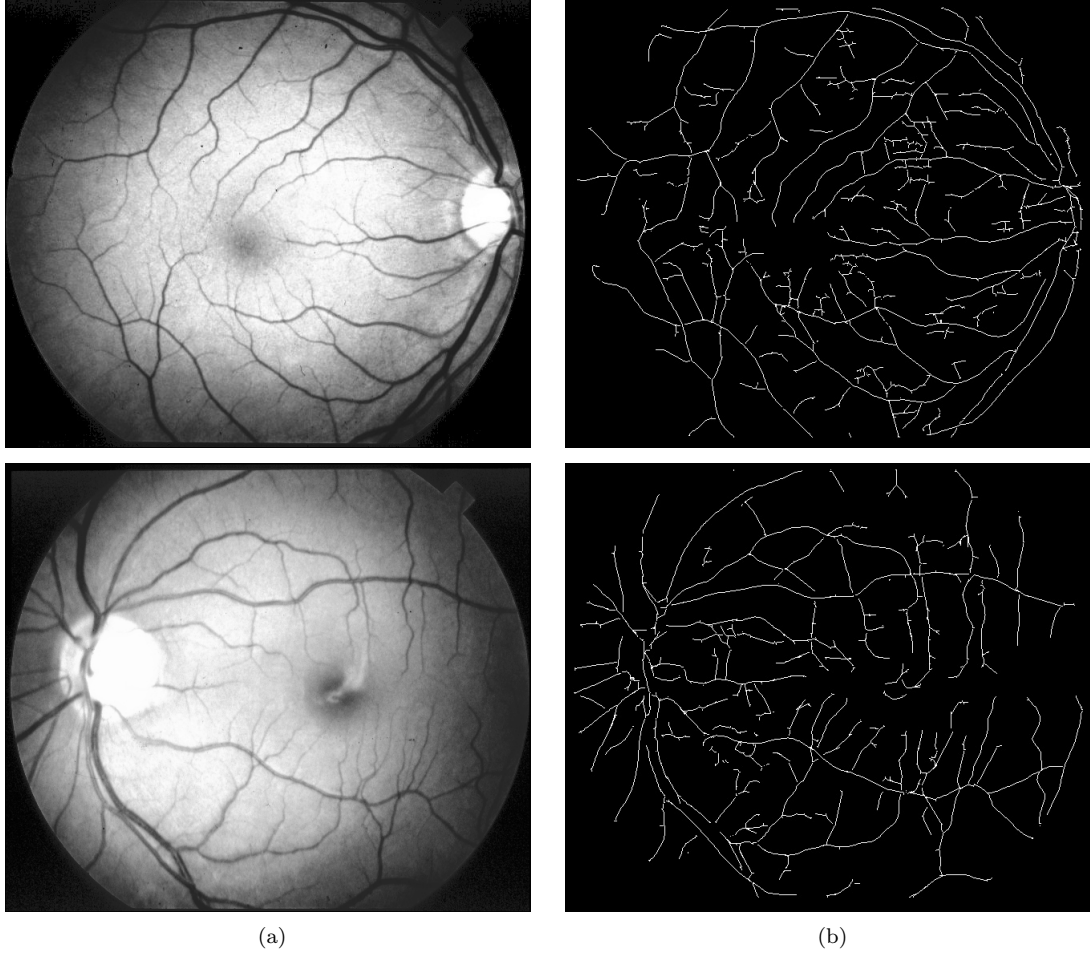


Figure 4.1. Skeletons of retinal images. From left to right, original MRA images and computed skeletons

Table 4.1. A table of the proposed method’s average processing time, in milliseconds, when applied to the images from the STARE database

Enhancing	Extracting	Building	Smoothing	Total
1329ms	203ms	152ms	345ms	2027ms

350 milliseconds. On average, the proposed method takes about 2000 milliseconds (or 2 seconds) to process a given image.

To assess the performance of the proposed method, we compare our results to the hand-labeled ground-truth vessel segmentations, provided by the STARE database, as well as the results of the piecewise threshold probing method [4]. In Figure 4.2, (a) and (b) shows the vessels hand-labeled by the first and second observer respectively, while (c) presents the vessel skeletons computed by the piecewise threshold probing method. In (d), the result from the proposed method is presented. Note that the first observer was aggressive in labeling the small vessels as shown

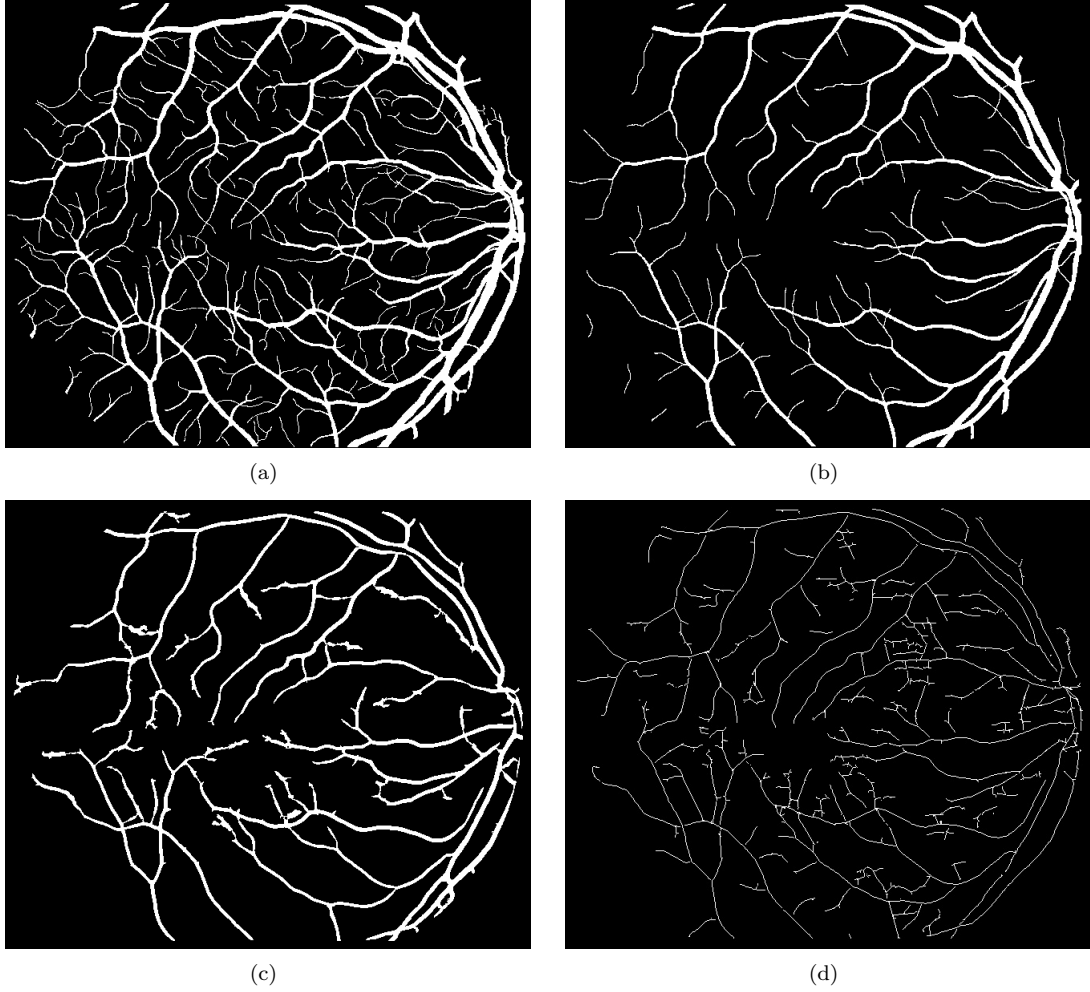


Figure 4.2. Comparison of results for a STARE image. (a) Hand-labeled vessels by the aggressive observer, (b) hand-labeled vessels by the non-aggressive observer with fewer small vessels identified, (c) the vessels by the piecewise threshold probing method, and (d) the vessels by the proposed method.

in (a), while the second observer was not aggressive and did not identify the small vessels as in (b). From the figure, we can see that the proposed method identified most of the vessels labeled by the aggressive observer, although it failed to identify the small vessels. The figure also shows that the proposed method identified far more vessels than the non-aggressive observer, as well as the piecewise threshold probing method.

Figure 4.3 shows similar results. The hand-labeled results of the aggressive and non-aggressive observer are presented in (a) and (b), respectively, while the result of the piecewise threshold probing method is presented in (c). The vessels identified by the proposed method are presented in (d). Similar to the previous comparison, most of the vessels identified by the aggressive observer are identified by the proposed method, but not all. However, the proposed method, again, identified far more vessels than both the non-aggressive observer and the piecewise threshold

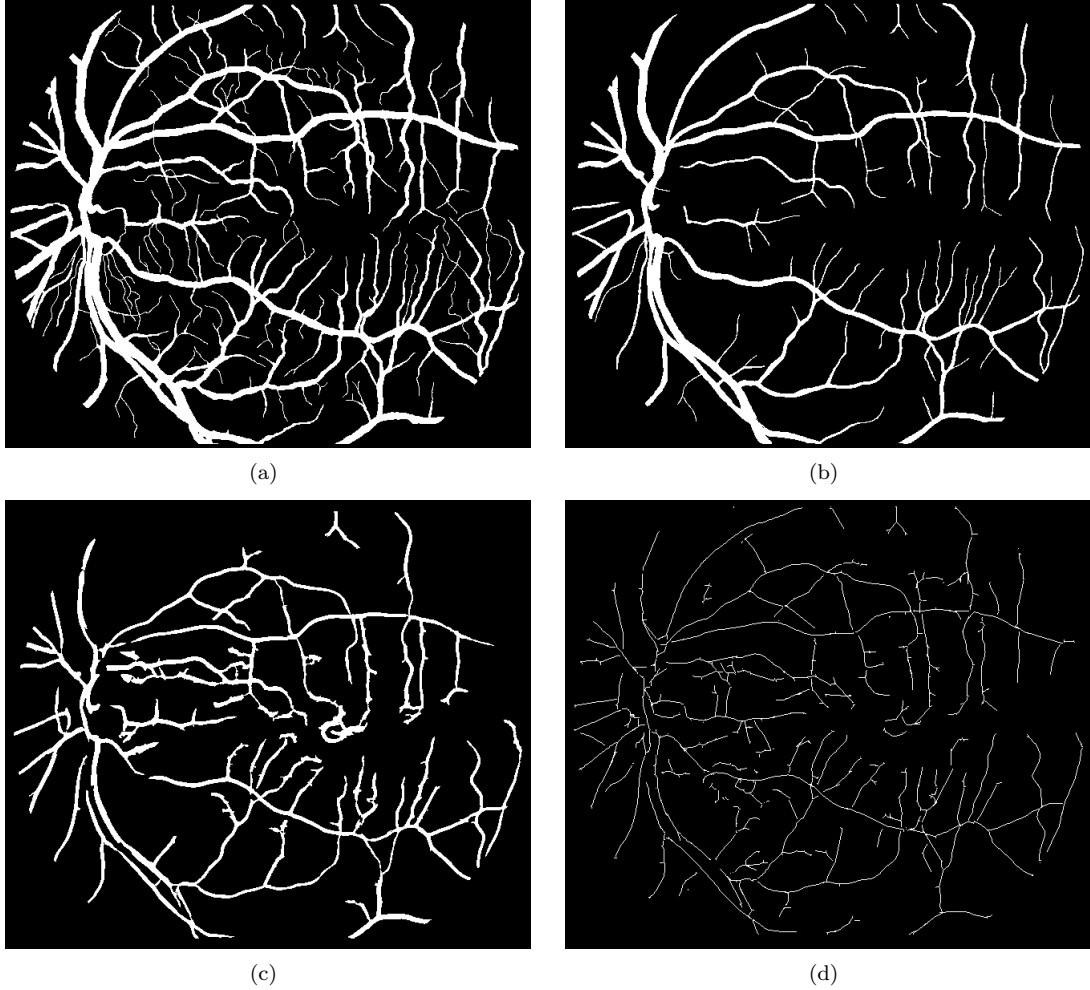


Figure 4.3. Comparison of results for a STARE image. (a) Hand-labeled vessels by the aggressive observer, (b) hand-labeled vessels by the non-aggressive observer with fewer small vessels identified, (c) the vessels by the piecewise threshold probing method, and (d) the vessels by the proposed method.

probing method. From these observations, we can say that the proposed method outperformed both the hand-labeled results by the non-aggressive observer and the piecewise threshold probing method on these two examples.

To evaluate the performance of the proposed method in the worst case, we applied the method to abnormal retinal images, i.e., those that contain pathology such as hemorrhages and microaneurysms. In Figure 4.4, the proposed method successfully detects and skeletonizes vessels outside of the abnormal region of the image. However, in the abnormal region in the image, the false-positive rate for vessel detection is higher, since the proposed method skeletonizes hemorrhages and microaneurysms as well.

We compared the results of our method applied to abnormal images to the results by the expert observers as well as the piecewise threshold probing method. It is important to note that



Figure 4.4. Skeletons of an abnormal retinal image. From left to right, original MRA image and computed skeletons.

even human observers had difficulty in accurately identifying vessels in and around pathology [4]. As illustrated in Figure 4.5, the proposed method detected more vessels than the non-aggressive observer as well as the piecewise threshold probing method outside of the abnormal region. However, in the abnormal region, our method produced a huge amount of false-positive vessels. This is due to the fact that our method is designed to skeletonize elongated objects by detecting ridges on a topological surface while the pathology creates false ridges on the surface. Further research is warranted to isolate the blob-like pathology in order to avoid skeletonizing false ridges [20].

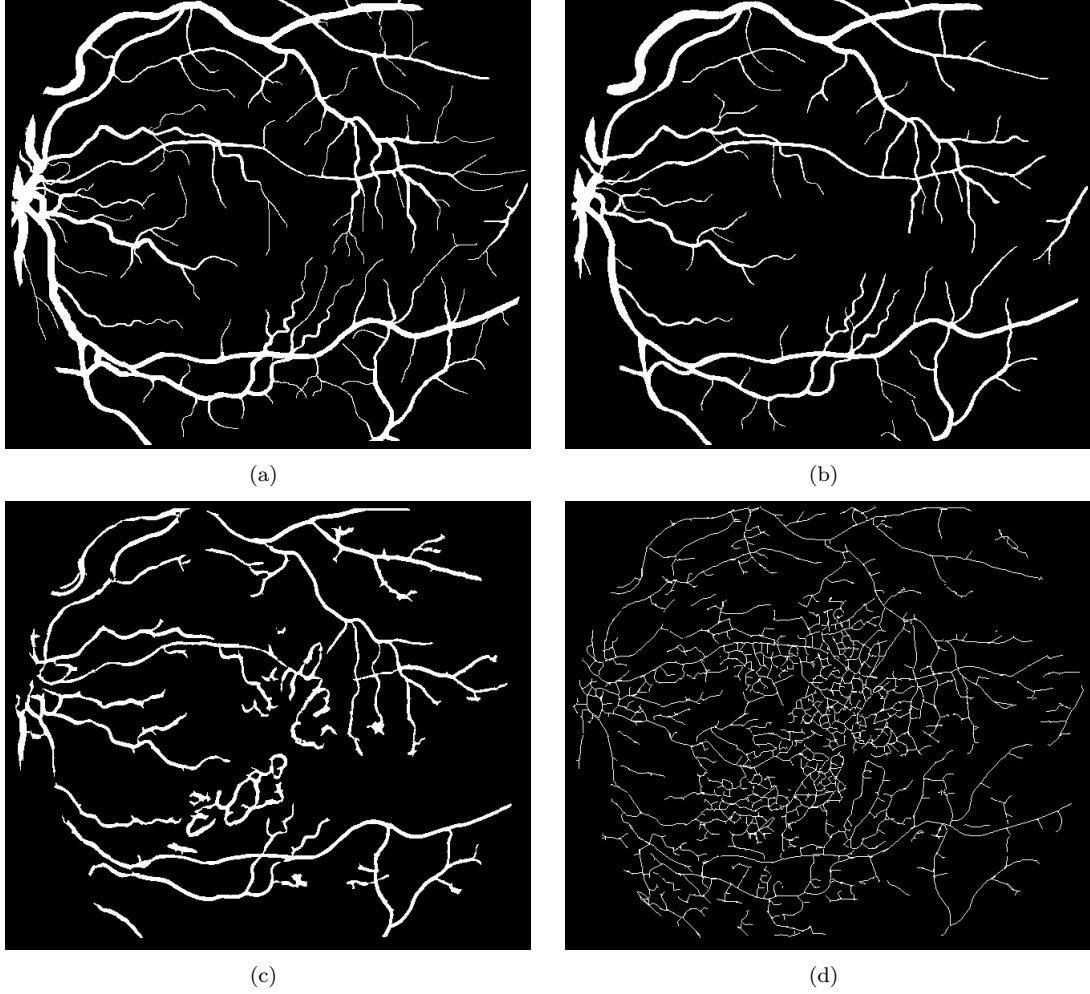


Figure 4.5. Comparison of results for an abnormal STARE image. (a) Hand-labeled vessels by the aggressive observer, (b) hand-labeled vessels by the non-aggressive observer with fewer small vessels identified, (c) the vessels by the piecewise threshold probing method, and (d) the vessels by the proposed method.

Conclusion

5.1 Summary

This thesis presents a novel and efficient skeletonization method that rapidly constructs the skeleton graphs in the discrete image space and, if desirable, obtains the smooth skeletons by applying the active contour models to the graphs. Matched filter is used to enhance vessels and reduce noise in the images. We define a set of gradient sign changes that can be used to identify ridge points present in grayscale images. The ridge points are extracted from the topological surfaces by detecting the gradient sign changes on two orthogonal scanlines. Then, recognizing that ridge-to-border intensity ratios are generally low for noise, the ratio filter method eliminates false-positive ridge points. The skeleton graph is constructed by linking the ridge points into a graph structure. Finally, the skeleton is converted to a set of active contour models, or snakes, and deformed according to the topology of the matched filter surface. The net result is a smooth and thin skeleton.

The proposed method differs significantly from the existing approaches in three respects. First, it does not require explicit information about the object boundaries or the critical points on them. Instead, it directly processes grayscale images. Second, it processes a given image as a whole, in contrast to most existing approaches where an individual object is a processing unit. As a result, the presence of multiple objects is automatically detected and the skeletons of the objects are computed simultaneously. Finally, since the skeletons are represented as the graph structures, we can easily obtain the various morphological parameters of the complex network structures.

We presented the promising results of the method applied to various retinal images. We showed that salient vessels, as well as less obvious vessels, were successfully detected and skeletonized. We analyzed the processing time of the proposed method, and showed that, on average, an image of 700×605 pixels can be skeletonized in about 2 seconds. Using both normal and abnormal images, we compared performance of the method to that of an aggressive and non-

aggressive human observer, as well as the piecewise threshold probing method. The proposed method outperformed the non-aggressive observer and the piecewise threshold probing method on all images.

5.2 Future Work

As of this writing, some weaknesses exist in the proposed method’s current implementation. Existing routines will be modified and new routines added to improve performance. There are five areas of focus that will lead to improvements of the method’s overall performance.

The first area of focus is on ratio filtering, which is designed to calculate the ratio between a ridge pixel and its nearest border pixel. The current ratio filter implementation seeks out the nearest object border pixel on the image’s horizontal or vertical vector. Instead, a sampling of the ridge segment orientation will be used to calculate a vector perpendicular to the object. The nearest border pixel on this vector, then, would be used for the ratio filter calculation.

The second area of focus is on the ordering of some routines. In the current implementation, the proposed method performs line patching followed by pruning. However, there are instances in which false-positive ridge segments are so close together that the line patching routine patches them together. The newly patched segment length then exceeds the pruning threshold and thus, the pruning method does not remove it from the graph structure. This introduces false-positive ridge segments into the skeleton. However, if pruning were to take place before line patching, small true-positive vessel segments would be eliminated from the skeleton. Consider a vessel disconnected at many points, resulting in very short ridge segments. If the pruning routine were to be applied before line patching, those short ridge segments would be eliminated, leaving gaps in the skeleton. As of the time of this writing, trial-and-error was the primary driver for deciding on the current order. More investigation is needed into the order of operations.

The third area of focus is on improving detection of small vessels. In its current state, the proposed method has difficulty in identifying some very small/fine vessels. From visual observation, a small vessel’s tortuosity is less pronounced than ridge segments created by noise. This suggests that a tortuous measurement routine may improve the proposed method’s ability to identify and eliminate noise.

The fourth area of focus is on connectivity. Connectivity suffers when abnormalities are present in an image. The proposed method is currently designed to skeletonize elongated objects, not blobs. New routines to detect and handle abnormalities will improve the overall performance.

The final area of focus is on pathology avoidance. Pathology is generally made up of blobs that cause false-positive ridges. Isolating and avoiding pathology will reduce false-positive ridges improving the overall performance of our method.

Beyond improving the method’s performance, there is also the matter of utilizing the resulting graph structure to measure the morphological parameters of vascular networks. For instance, bifurcation points can be detected by scanning the graph for any point where two or more deformable curves meet. Measuring the curvature of a branch at some point can be accomplished

by taking the reciprocal of the radius of a circle that most closely approximates the curve at that point. Since branches are open, length can be measured by calculating the arc length of the curve. Branch width, at some point on the skeleton, can be measured by calculating the distance between that point's two nearest boundary points.

Bibliography

- [1] WESARG, S., M. KHAN, and E. FIRLE (2006) “Localizing Calcifications in Cardiac CT Data Sets Using a New Vessel Segmentation Approach,” *Journal of Digital Imaging*, **19**, pp. 249–257.
- [2] JIANG, X. and D. MOJON (2001) “Blood Vessel Detection in Retinal Images by Shape-Based Multi-Threshold Probing,” in *Pattern Recognition* (B. Radig and S. Florczyk, eds.), vol. 2191 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 38–44.
- [3] AL-DIRI, B. and A. HUNTER (2009) “Automated Measurements of Retinal Bifurcations,” in *World Congress on Medical Physics and Biomedical Engineering, September 7 - 12, 2009, Munich, Germany* (R. Magjarevic, O. Dssel, and W. C. Schlegel, eds.), vol. 25/11 of *IFMBE Proceedings*, Springer Berlin Heidelberg, pp. 205–208.
- [4] HOOVER, A., V. KOUZNETSOVA, and M. GOLDBAUM (2000) “Locating Blood Vessels in Retinal Images by Piecewise Threshold Probing of a Matched Filter Response,” *IEEE Transactions on Medical Imaging*, **19**(3), pp. 203–210.
- [5] NIEMEIJER, M., J. STAAL, B. VAN GINNEKEN, M. LOOG, and M. ABRAMOFF (2004) “Comparative study of retinal vessel segmentation methods on a new publicly available database,” in *SPIE Medical Imaging* (J. M. Fitzpatrick and M. Sonka, eds.), vol. 5370, SPIE, SPIE, pp. 648–656.
- [6] BLUM, H. (1973) “Biological Shape and Visual Science (Part 1),” *Journal of Theoretical Biology*, **38**, pp. 205–287.
- [7] YIM, P., P. CHOYKE, and R. SUMMERS (2000) “Gray-Scale Skeletonization of Small Vessels in Magnetic Resonance Angiography,” *IEEE Transactions on Medical Imaging*, **19**(6), pp. 568–576.
- [8] BARUH, H., F. LEYMARIE, and M. LEVINE (1992) “Simulating the Grassfire Transform Using an Active Contour Model,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**(1), pp. 56–75.
- [9] OGNIWICZ, R. L. and O. KUBLER (1995) “Hierarchic Voronoi Skeletons,” .
- [10] MONTANARI, U. (1969) “Continuous Skeletons from Digitized Images,” *Journal of ACM*, **16**(4), pp. 534–549.
- [11] MARTINEZ-PEREZ, M., J. J., and J. NAVALON (1987) “A Thinning Algorithm Based on Contours,” *Computer Vision, Graphics, and Image Processing*, **39**, pp. 186–201.

- [12] ARCELLI, C. and G. SANNITI DI BAJA (1985) “A Width-Independent Fast Thinning Algorithm,” *IEEE Transactions on Pattern Recognition and Machine Intelligence*, **7**(4), pp. 463–474.
- [13] AYLWARD, S. and E. BULLITT (2002) “Initialization, Noise, Singularities, and Scale in Height Ridge Traversal for Tubular Object Centerline Extraction,” *IEEE Transactions on Medical Imaging*, **21**(2), pp. 61–75.
- [14] MCINTOSH, C. and G. HAMARNEH (2006) “Vessel Crawlers: 3D Physically-based Deformable Organisms for Vasculature Segmentation and Analysis,” in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, pp. 1084 – 1091.
- [15] CHAUDHURI, S., S. CHATTERJEE, N. KATZ, M. NELSON, and M. GOLDBAUM (1989) “Detection of Blood Vessels in Retinal Images Using Two-Dimensional Matched Filters,” *IEEE Transactions on Medical Imaging*, **8**(3), pp. 263–269.
- [16] KASS, M., A. WITKIN, and D. TERZOPOULOS (1988) “Snake: Active Contour Models,” *International Journal of Computer Vision*, **1**(4), pp. 321–331.
- [17] WILLIAMS, D. J. and M. SHAH (1992) “A fast algorithm for active contours and curvature estimation,” *CVGIP: Image Underst.*, **55**, pp. 14–26.
- [18] BARUH, H. (1999) *Analytic Dynamics*, McGraw-Hill.
- [19] TERZOPOULOS, D. (1983) “The Role of Constraints and Discontinuities in Visible-Surface Reconstruction,” in *Proceedings of IJCAI*, pp. 1073–1077.
- [20] CHANG, S., H. EMOTO, D. METAXAS, and L. AXEL (2004) “Pulmonary Micronodule Detection from 3D Chest CT,” in *Medical Image Computing and Computer-Assisted Intervention MICCAI 2004* (C. Barillot, D. Haynor, and P. Hellier, eds.), vol. 3217 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 821–828.