

Adaptive Segmentation of Cardiovascular Vessels

A Thesis
by
ERIK JOHN COLE

Submitted to the Graduate School
at Appalachian State University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

December 2019
Department of Computer Science

Adaptive Segmentation of Cardiovascular Vessels

A Thesis
by
ERIK JOHN COLE
December 2019

APPROVED BY:

Dr. Rahman Tashakkori
Chairperson, Thesis Committee

Dr. James B. Fenwick Jr.
Member, Thesis Committee

Dr. Cindy A. Norris
Member, Thesis Committee

Dr. Rahman Tashakkori
Chairperson, Department of Computer Science

Dr. Michael McKenzie, Ph.D.
Dean, Cratis D. Williams School of Graduate Studies

Copyright© Erik J Cole 2019
All Rights Reserved

Abstract

Adaptive Segmentation of Cardiovascular Vessels

Erik J Cole

B.S.E., University of Iowa

M.S., Appalachian State University

Chairperson: Dr. Rahman Tashakkori

Coronary collateral vessels are small vessels that may contribute to survival after myocardial infarction by providing blood to the cardiac muscle after coronary arterial occlusion. These collateral vessels develop in part due to increasing pressure caused by an occluded vessel helping to provide blood to otherwise inaccessible tissue. However, these vessels are not present in all people and can develop after infarction and in some cases they develop prior to infarction for reasons not fully understood.

The goal of this thesis is to investigate the segmentation of coronary collateral vessels from micro-computed tomography (microCT) images of a mouse's heart. A problem limiting study of collateral vessels is exceedingly small size and correspondingly low blood flow of these vessels, making the regions of interest (ROI) below the resolution of most imaging modalities. Segmentation of vessels is a challenge for all imaging modalities and organs. There is no standard algorithm or method that works for all images, therefore, a combination of multiple approaches were used to address this problem. This thesis found utilizing Sobel and Gaussian Filters, Regional filtering, and a global threshold to produce

best results for segmenting the coronary collateral vessels in the microCT images of a mouse's heart.

ACKNOWLEDGEMENTS

I give a heartfelt thank you to my mentor, Dr. Rahman Tashakkori who has provided me honest feedback, guidance, and support throughout my time at Appalachian State University. This medical imaging project would not have been possible without his support.

I would also like to thank my committee members, Dr. Fenwick and Dr. Norris for their time and effort. Their contributions are highly appreciated. My professors here at Appalachian State University have helped equip me with the skills needed to undertake this project, and I thank them.

Finally, thank you to my family who has supported me throughout my academic years. Without their support I would have never made it this far. Additionally, to all my friends here at Appalachian State University, thank you for helping to make my time here enjoyable. Your presence contributed to my success. I hope to see you all again.

Contents

Abstract	iv
Acknowledgements	vi
List of Figures	ix
1 Introduction	1
2 Theoretical Background	3
2.1 Physiological Background	4
2.2 Thresholding	6
2.3 Region Growing	8
2.4 Hessian Filter	9
2.5 Level Sets	11
2.6 Sobel Filter	12
2.7 Gaussian Blur Filter	13
2.8 Interpolation	13
3 Literature Review	15
4 Methodology	20
4.1 Data Construction	21
4.2 Pre-processing	23
4.3 Filtering	24
4.4 Regional Filtering	26
4.5 Region Growing	27
4.6 Interpolation	29
4.7 3D Display	31
5 Results	32
5.1 Preprocessing	32
5.2 Regional Results	34
5.3 Threshold Results	36
5.4 Interpolation Results	36
5.5 Filter Results	41

6 Conclusion and Future Work	44
6.1 Preprocessing	44
6.2 Regional Filter	44
6.3 Sobel and Gaussian Filtering	45
6.4 Interpolation and Region Growing	45
6.5 Future Work	46
Bibliography	48
Vita	50

List of Figures

4.1	The Flow Diagram of the Research	21
4.2	Slices and trimmed slice	22
4.3	Slice Stacking	23
4.4	Heart Visualization with no processing	25
4.5	Heart Visualization with a Threshold	26
4.6	Heart Visualization with a Discontinuity	28
4.7	Search pattern for subsequent slices	30
5.1	Example of background removal	33
5.2	Regional Filtering: a-i - original images and a1-i1 - their corresponding results	35
5.3	Results of Global Threshold	37
5.4	Example of Interpolated Missing Slice	38
5.5	Example of Interpolation between Slices	40
5.6	Display before filtering	42
5.7	Display after filtering	43

Chapter 1

Introduction

A myocardial infarction (Heart attack) is a serious medical event that affects about 735,000 people in the United States every year. According to the Centers for Disease Control and Prevention (CDC), over 500,000 of these patients are experiencing their first heart attack, while over 200,000 experience a second heart attack [1]. People survive a heart attack for many different reasons, however it is not always possible to predict survival or determine the reason for survival for a significant number of cases. Some suggested heart attack preventative measures are aimed at lifestyle practices: regular exercise, a better diet, managing weight, and not smoking. While these lifestyle practices have been proven effective, the etiology of heart attacks and reasons for surviving a heart attack still are not always known. One known effect of exercise is that it increases coronary collateral vessel growth on the heart which may contribute to survival [13]. These vessels are the focus of the research in this thesis.

Coronary collateral vessels can provide redundant blood supply to heart tissue if a blockage occurs in the main pathways. These vessels are extremely small and challenging to locate in medical images. The research described here aims to identify these collateral vessels from microCT. The results reported here may be valuable to further physiological studies tracking the presence of and change in these vessels or to assess a patient's chance of survival in case of a heart attack. Using an adaptive local threshold, Sobel and Gaussian filters, and other refinement techniques the vessels are able to be segmented and displayed.

This thesis is organized as follows: Chapter 2 - Theoretical Background provides a summary of the theories used in this research. Chapter 3 - Literature Review provides a review of existing work and Chapter 4 - Methodology addresses how selected theories are implemented in this thesis. Chapter 5 - Results provides some of the results and Chapter 6 - Conclusion and Future Work provides a summary of findings and highlights future work.

Chapter 2

Theoretical Background

Segmentation refers to the processes of drawing a boundary around a region of interest. This can be done by hand but requires training and time. Image segmentation is a complex problem and currently there is no single algorithm that works efficiently for all imaging modalities and subjects. Medical imaging has many different parameters and image types which present a broad spectrum of images. Two main categories exist for image segmentation: region-based and edge-based. The region-based methods include thresholding, clustering, and region growing and the edge based methods include machine learning, fuzzy logic, and genetic algorithms. Edge-based methods utilize edge detecting algorithms such as Sobel, Prewitt, or Canny [15]. Many methods can be used to achieve comparable results across various categories depending on the goal of the segmentation [10]. Rudyanto et al. compared 23 different methods from 17 different teams. These methods comprised of primarily four sub-categories: thresholding, region growing-based,

machine-learning, and Hessian-based. These four methods can be combined, in some cases, to create a hybrid method. These methods are discussed later in this chapter.

2.1 Physiological Background

The heart is a muscle that requires blood to provide it oxygen and nutrients so that it can pump blood to the rest of the body. The heart is therefore innervated by vessels on the outside. Cardiac muscle tissues receive their main supply of blood from the two coronary arteries that branch off the aorta. Having two coronary arteries provides redundancy to one of the most critical organs in the body. If one becomes partially blocked, then the other can still pump blood to the heart. However, a full blockage of a coronary artery can lead to heart attack or other serious issues. The human body's responses to partially occluded arteries includes chest pain, shortness of breath, and potentially the growth of collateral vessels.

Collateral circulation refers to a situation where an artery or vein is occluded but blood is still able to circulate through nearby vessels in the affected area. These alternative routes for blood flow are sometimes redundant pathways already grown by the body to maintain blood supply to a critical region. However, in some cases, new vessels are grown in response to a stimulus such as a blocked vessel. Unfortunately, some factors, such as diabetes, can inhibit the formation of collateral vessels [13]. The existence of coronary collateral vessels is crucial to surviving a myocardial infarction (heart attack) caused by a blocked coronary artery.

Collateral vessels grow and form as a result of an increase in vascular pressure [2]. Chilian et al. and Tatli et al. both evaluated coronary collateral flow indirectly by testing blood flow [2] or by evaluating the amount of blockage present in a larger artery [13]. Coronary collateral vessels are at most 50 microns in diameter [12]. Investigating the formation or presence of coronary collateral vessels indirectly through other factors or measurements is not precise and limits the accuracy of measurements. However, due to the small scale of these vessels most medical imaging modalities are not suitable to accurately and reliably capture coronary collateral vessels.

A microCT image usually contains many files in Digital Imaging and Communications in Medicine format (DICOM). DICOM format consists of a header with useful information about the image acquisition device, scan parameters, subject's information, image parameters, other relevant information, and most importantly, the pixel array representing the image itself. The pixel array contains the image data and when all pixel arrays are combined a three dimensional volume can be created. The pixel array is usually 8-bit grayscale but in case of a microCT image a 16-bit format is not uncommon.

X-ray microcomputed tomography (microCT) can achieve an image resolution of 3 to 50 microns [9]. A higher image resolution of a scan results in smaller objects being captured. MicroCT, one of the highest resolution imaging modalities, is therefore viable for successfully capturing coronary collateral vessels. MicroCT is a method where many x-ray images are taken at varying depths. This imaging modality therefore creates many two-dimensional slices of a three-dimensional object. The resulting slices can be used to recreate the three-dimensional objects including some internal features. For example, a

heart containing empty chambers could be scanned and the 3D recreation would contain the empty internal chambers. With microCT, the resolution able to acquire some of collateral vessels from the surrounding tissue.

High-resolution microCT scans allow for the segmentation of the blood vessels. Segmentation refers to the process of partitioning an image by selecting objects or regions of interest by a user. For example, selecting a tumor in a medical image, a person in a forest, or a building in a skyline. Segmentation can be done manually through tracing regions of interest on an image, however, this process is time intensive and prone to bias through individual determinations. Ideally, automated segmentation would be used to remove the human bias from the process. Methods for image segmentation are discussed later in this thesis.

In this thesis, image segmentation is used to create the initial skeleton of the blood vessels in a set of microCT images of a mouse's heart. This segmentation utilizes thresholding, filtering background and cardiovascular tissue, refinement of vessel structure, and error correction methods. These are described in Chapter 4.

2.2 Thresholding

Thresholding is a straightforward approach that is widely used in image segmentation. While thresholding can be done on a subsection of an image as shown by Jiang and Mojon [3], it generally refers to a global segmentation in which a threshold is applied to the whole image. Thresholding applied on a subsection of an image will be referred to as

local thresholding to prevent confusion. Thresholding can be implemented by choosing a target intensity value and then checking every pixel on an image to investigate whether it is greater than or equal to the intensity. In this process, pixel values meeting the above criteria can be set to a fixed value, such as 1, and pixel not meeting the criteria to another fixed value such as 0. This would result in a binary image being created. In such a case, the resulting binary values can be stored to create a binary matrix of the same size as the image, and this matrix can be used as a mask to retain only the regions of interest on the image (i.e. the regions that meet the threshold criteria). This type of thresholding works best when a clear contrast is apparent between the region of interest and the rest of the image. For example, when bone surrounded by soft tissue is the region of interest on an x-ray image, it would be an ideal time to use a thresholding approach to separate the pixels representing the bone from those of the tissue.

Thresholding has more complex options besides a simple threshold value discussed above. Iterative thresholding is a viable approach when no clear distinction is apparent or if multiple regions of interest could exist on the image [5]. Iterative thresholding allows for the discovery of a threshold without a prior knowledge of a suitable value on the image but requires a more robust evaluation method to provide feedback to the algorithm. Other threshold determinations such as Otsu's method, maximum correlation, clustering, or adaptive thresholding exist but are not relevant to the images used in this thesis [5, 15].

Another interesting thresholding method is adaptive thresholding [5]. The application of repeated thresholds of non-overlapping ranges allows for multiple objects to be

segmented. Due to blood vessels of varying sizes having different intensities in microCT images, adaptive thresholding is a better approach for segmenting vessels of differing diameter. The pixel intensity of blood vessels decreases relative to the diameter as less air/fluid is inside the vessel. The attenuation of the vessel trends toward that of the surrounding tissue, which makes identification of tiny vessels challenging on lower resolution images.

Edge-based thresholding methods exist but are not suitable for this type of segmentation. Edge detection struggles to find all but the largest of vessel edges due to the overlapping vessels and the complex nature of the vessels. Applying an edge detection algorithm to the images results in many fragmented edges that do not provide context or relevant information.

2.3 Region Growing

Region growing starts with a single pixel (usually selected manually) then continues by including similar ones from its neighboring pixels. Similarity can be measured by pixel intensity, pixel variance, correlation, texture, or shape. Region growing guarantees a connected region from the seed; however, if multiple seeds are used, unconnected regions may occur. In images with less noise or with noise filtered out, region growing can provide fast results with minimal error. However, when the image is noisy, or the region of interest contains large changes in its pixel intensity, then this method results in many smaller regions that may not be relevant.

Seed pixel selection is crucial when performing region growing. Manual seed selection is common in this approach, but automated selection can be done as well. The selection of a seed must be consistent as different seeds can change the segmented region. Furthermore, if multiple seed pixels are chosen, the computational time needed to complete the segmentation can increase with no clear benefit. Choosing multiple seed points can prevent problems with disconnected regions at a high cost of computational time and work. Multiple overlapping seeds results in an unsatisfactory waste of system resources.

Coronary vessels are connected and form trees with the left and right coronary arteries acting as the roots of the trees. Region growing therefore appears to be a suitable approach for segmenting the vessels while retaining their spatial orientation. Additionally, because vessels get smaller as they get farther away from arteries that feed them, the small collateral vessels of interest should be present at the end of the region growing process. If collateral vessels are present, then the left and right arterial vessel trees should be connected to each other resulting in one tree.

2.4 Hessian Filter

Like edge detection algorithms that look for edges by using contrasting pixel intensities, the Hessian filter can search for curves existing in a 2D image. By applying a Hessian filter, the curvature of objects such as, blood vessels, can be studied in 2D images. When applied to the microCT images used in this thesis, the Hessian filter vessels are isolated

due to the nature of the blood vessels presenting with non-linear trends. The application of a Hessian filter can help remove non-vessels items. The Hessian filter is based off the Hessian matrix. This matrix consists of second order partial derivatives. The application of a Hessian filter requires a convolution with a Gaussian resulting in a blurring of edges [4]. However, this blurring can be used with region growing methods to solve some of the disconnects in the vessels on the image.

Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a function taking as input a vector $x \in \mathbb{R}^n$ and outputting a scalar $f(x) \in \mathbb{R}$; if all second partial derivatives of f exist and are continuous over the domain of the function, then the Hessian matrix \mathbb{H} of f is a square n -by- n matrix, usually defined and arranged as follows:

$$\mathbb{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \quad (2.1)$$

This can be stated as an equation for the coefficients using indices i and j :

$$\mathbf{H}_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

Using a Hessian matrix to isolate the pixels that most likely contain vessels reduces error and aids in segmentation by eliminating things that do not represent vessels. However, this filter must be applied to every pixel on each 2D image, This application of the Hessian filter to a microCT image is computationally expensive.

2.5 Level Sets

Image segmentation is about identifying boundaries around regions of interest. This can be done manually, tracing the borders of region by hand or can be automated with some risk of accuracy loss and completeness. However for large images, manual segmentation can consume a significant amount of time and still contain some error. One method of automatic segmentation uses level sets [8]. Osher and Sethian first proposed using level sets to enclose a surface.

Level sets is a method of segmentation that works similar to region growing but also keeps track of the contour level of the feature. This method can track the vessels as they split and merge while also maintaining information about the vessels location and size.

Malladi proposed using this in medical images and demonstrated its success [6]. By propagating a 2D surface through the 3D data a complex curve can be created. In the case of blood vessels in our images, due to the preprocessing, the vessels are at a single contour level while the surrounding tissue is set to another. Using level sets, only one level will need to be enclosed as the vessel level. Level sets are a form of region

growing that is constrained by the distance from a seed point or the contour currently being examined. This means the constraints on region growing methods generally apply to level sets.

2.6 Sobel Filter

The Sobel filter is an edge detection filter. Edges are detected by calculating the gradient along both x and y axes by convolving the image with each of these two matrices, respectively.

$$\text{X-axis matrix: } \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

$$\text{Y-axis matrix: } \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

These matrices allow for a resulting value to indicate where a potential edge is by finding places with larger changes in the perceived gradient of the image data. This filter is not ideal for noisy images or images that lack consistent features.

2.7 Gaussian Blur Filter

The Gaussian blur filter can reduce noise in an image by smoothing the image. The smoothing of the image is achieved by representing every pixel value with the average of the pixels in a region of specific size surrounding that pixel. A Gaussian filter is a low-pass filter in which the higher value pixels are reduced in intensity and the lower value ones are increased. The reduction in extreme values reduces image noise and is a valuable preprocessing step in which edges are blurred and the contrast is reduced.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.2)$$

Where x represents the pixel distance from the origin in the horizontal axis, y represents the the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution. When this filter is applied on an image, it produces a surface whose contours are concentric circles with a Gaussian distribution from the center point. Equation (2.2) is used to create a matrix containing a value for every pixel. The resulting values then convolved with the original image producing a smoothing effect known as Gaussian Blur.

2.8 Interpolation

Interpolation is the ability to find missing values at some data points based on the already known values at the neighboring data points. The new data point is an approximation using the surrounding data points to provide insight into the trend of the data at that

point. Interpolation utilizes a mathematical formula that best describes the trend of the existing data. One of the most common interpolation techniques is linear interpolation. In this technique the assumption is that the behavior of the points follow a linear equation defined by equation (2.3) where m is the slope of the line and b is the y intercept. For an image the slope is gradient between two points. Therefore to interpolate with a linear technique, after calculating the gradient it is simple to find the value of the interpolated point by substituting the x and y coordinates into equation (2.3).

$$y = mx + b \tag{2.3}$$

Spline interpolation is a more complex form of interpolation than linear in which instead of finding a single resulting formula, a low order polynomial is generated for each interval between the known data points. This results in a smaller error than linear interpolation when the trend of the data is not linear and is also computationally more efficient than using higher order polynomials to fit a larger curve.

Chapter 3

Literature Review

Little work has been aimed at identifying and segmenting coronary collateral vessels. Blood vessel segmentation for different organs including the heart has been studied before and studies of vessel segmentation have been conducted using various techniques. Validation of segmentation is done by comparing the results obtained from an automated program to those obtained manually. A manual segmentation takes time and can introduce error due to the human element. Machine learning, region growing, global thresholding, and adaptive local thresholding are all effective methods used by several efforts to segment blood vessels.

Tatli et al. [13] investigated the identification and segmentation of blood vessels present in a CT scan of a heart. Their paper covered the formation and development of the coronary collateral vessels and the significance of being able to research them further[13]. They demonstrated that coronary collateral vessels are relevant to surviving a myocardial infarction. Collateral vessel size is very small, approximately 200 microns.

The formation and factors promoting growth of coronary collateral vessels is also discussed with comparisons to the data providing some conclusions.

Tatli et al. classified the patients by the amount of blockage (without functional impact to the heart), but did not directly evaluate the state of the coronary collateral vessels[13]. It appears that they were not able to image or evaluate the coronary collateral vessels due to challenges in acquiring such small features. Instead, the authors relied on the state of the larger vessels to deduce the state of the coronary vessels. Finally, the authors conclude that further research on coronary collateral vessels is needed[13].

Rudyanto et al. investigated vessel segmentation in CT scans of the lungs as approached in a contest. An open call was put out for teams to contribute their algorithms for segmenting pulmonary vessels [10]. The control data were provided by medical students doing a manual segmentation. Manual segmentation took a significant amount of time for a single scan and was not feasible for large number of images. However, it provided a baseline to which the results were compared [10]. There were 20 teams that provided results. The results took the form of a pixel array containing either a binary value (vessel or not vessel) or an 8-bit number representing the probability that the pixel was a vessel. Hessian filters were a popular choice for the teams that submitted their results. Other methods included fuzzy algorithms, seed point algorithms, machine learning, and thresholding [10]. The various algorithms describe by Rudyanto et al. demonstrate the current approaches to vessel segmentation. Adaptation and tuning of these methods to apply to the cardiovascular system could provide better results and allow for more detailed analysis. The initial attempt at only thresholding and enhancing the image

struggle across different image sets with different parameters. However, using a more advanced algorithm should allow it to be applicable to most image sets [10].

Marks et al. attempted to quantify the volume of blood vessels from MRI or Micro-CT scans [7]. They noted the time intensive nature of the task and wanted a faster automated approach to acquire the measurements. Using ImageJ and Fiji, they developed a plug-in that was capable of achieving this. 3D skeletonization techniques were used to determine vessel paths on a thresholded image. They compared their results to commercial software and achieved a high accuracy. Furthermore, the reduced need of manual control resulted in a less laborious method than other options [7]. Their method used a thresholded image, but didn't require some manual selection of vessels of interest. The threshold value does not change for a set of scanning parameters however, meaning each set of scanning parameters requires some manual fine tuning for optimal performance. After the image is skeletonized, a tree is created for vessels allowing them to be correlated based on their connecting portions. After the tree is created the output is a 3D representation of the vessels for users to select or manipulate. The algorithm proposed in [7] seems applicable to cardiac collateral vessels due to the tree-like nature of vessel perfusion. Their approach is to threshold the image, create a skeletonized version of it, and then select vessels of further interest from an 3D reconstruction. This approach demonstrates a simple method to visualize and acquire information about vessels that are part of a larger system while retaining the spatial and relational data [7].

Schaap et al. [11] presented a learning method based on the geometry of the vessels. The training data consisted of manually annotated images. This machine learning

approach applied linear regression and nonlinear regression to identify vessels in a region of interest (ROI). They compared their results to a publicly available data set called the Rotterdam Coronary Artery Evaluation Framework. This showed positive results and demonstrates how vessel shape can contribute to their identification or segmentation. Additionally, they extracted center lines, which would help in creating a tree based network for an image [11].

Zaitoun and Aqel [15] investigated different segmentation techniques that are in use currently. Along with Rudyanto et al. a significant overview of the available image segmentation techniques are presented. Thresholding, region growing, machine learning, and filters (Hessian and others) are of particular interest to us. Zaitoun and Aqel classify segmentation as region based or edge based. These categories provide a basis for attempting to segment an image and introduce some methods previously not considered. While not going into much detail, Zaitoun and Aqel are able to discuss the benefits of many approaches along with the drawbacks. Their publication served as a strong overview of the available methods [15].

Yang [14] presented a method of cardiovascular vessel segmentation based on an active contour model using level sets. The goal was to identify coronary atherosclerosis from computed tomography angiography. This requires a segmented vessel tree that can then be searched for atherosclerosis plaque formations. Level sets first introduced by Osher and Sethian [8] assisted in providing a segmented vessel tree for further analysis. This work proves that applying a level set approach to cardiovascular segmentation is successful.

Adaptive Local thresholding is a technique presented by Jiang and Mojon [3]. Their work on segmenting the retinal vessels demonstrates the improved accuracy of an adaptive local threshold over a global threshold. They demonstrate that dividing an image that contains objects of interest varying in color or intensity or are non-uniformly sampled during image acquisition may require something more complex than a global threshold. By dividing the image into smaller regions then acquiring a threshold based on those regions a more complete segmentation can be accomplished. The drawback to this method is the increased computational time over a global threshold.

Vessel segmentation remains a complex task. Many approaches to segmentation exist which depend on various criteria including imaging modality, image resolution, physiological abnormalities, and physiological features. The existing methods currently rely on knowing specific physiological features of the vessels and surrounding tissue (ex. Tumor growth or abnormal tissue density). For the purpose of the research in this thesis, a mixed method is utilized that includes these critical parts: thresholding, an adaptive local threshold, and filtering. Both global and local thresholds are required along with Sobel, Guassian, and Edge Detection filters to refine the images before displaying them. The combination of these methods provided the best results, which are discussed in chapter 5.

Chapter 4

Methodology

This thesis utilizes images of mouse hearts provided by the UNC Medical Center. They were acquired using a microCT machine which has more accurate resolution compared to a normal CT machine. This resolution allows for the capture of smaller structures. The hearts were imaged after being removed from the mice. The only treatment to them involved having excess blood drained and the heart cleaned up to allow for a clean image capture.

Figure 4.1 shows the work-flow used in this thesis. After the regional filtering, there are 3 options to help refine the image being displayed. All 3 options are not required for display but were tested in this thesis.

Using a Python library a DICOM image can be loaded into memory. The total image size is approximately 8GB. This image size presents a problem on some machines that do not have sufficient RAM to allocate 8GB in contiguous fashion. The DICOM file must be specified in the code by giving the directory name containing all of the image

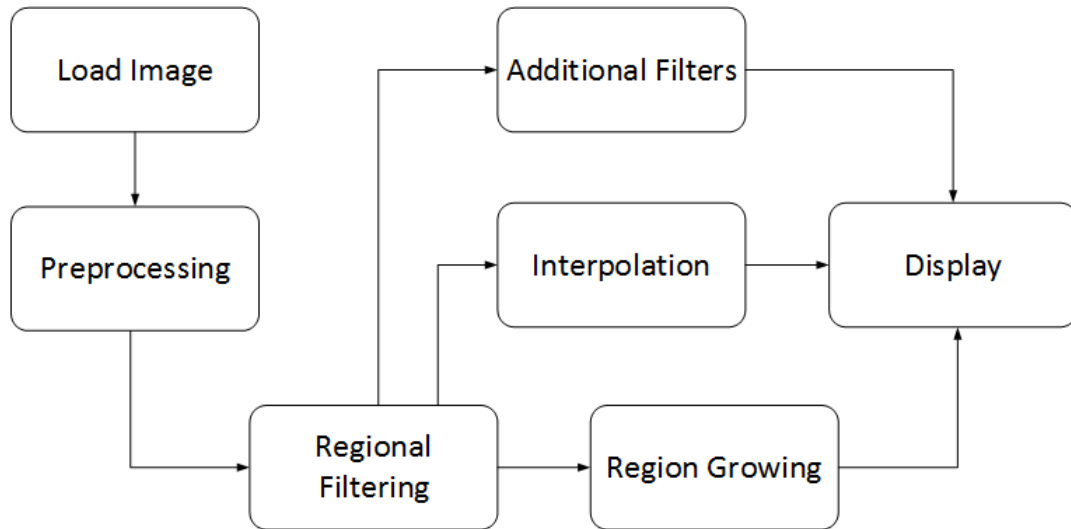


Figure 4.1: The Flow Diagram of the Research

slices. The image slices contain 16 bit pixel values. These values are stored in a three dimensional array, created by stacking the slices in order. Figure 4.2 shows the data from all 3 dimensions. The lower right image shows a cropped view of the scan. Removing the portion of the image that doesn't contain the heart reduces the memory required by 75% and reduces the runtime. The top scans show slices 250 and 700. Those scans are displayed along their original axis.

4.1 Data Construction

The first step in processing the image file is to obtain the image data from the DICOM file. The *pydicom* package in Python is used to read the DICOM file. The DICOM image format contains an array of pixel values and a header containing scan specifications, optional patient information, and machine details. The header provides the scan resolution and slice thickness among other important information. To load the data,

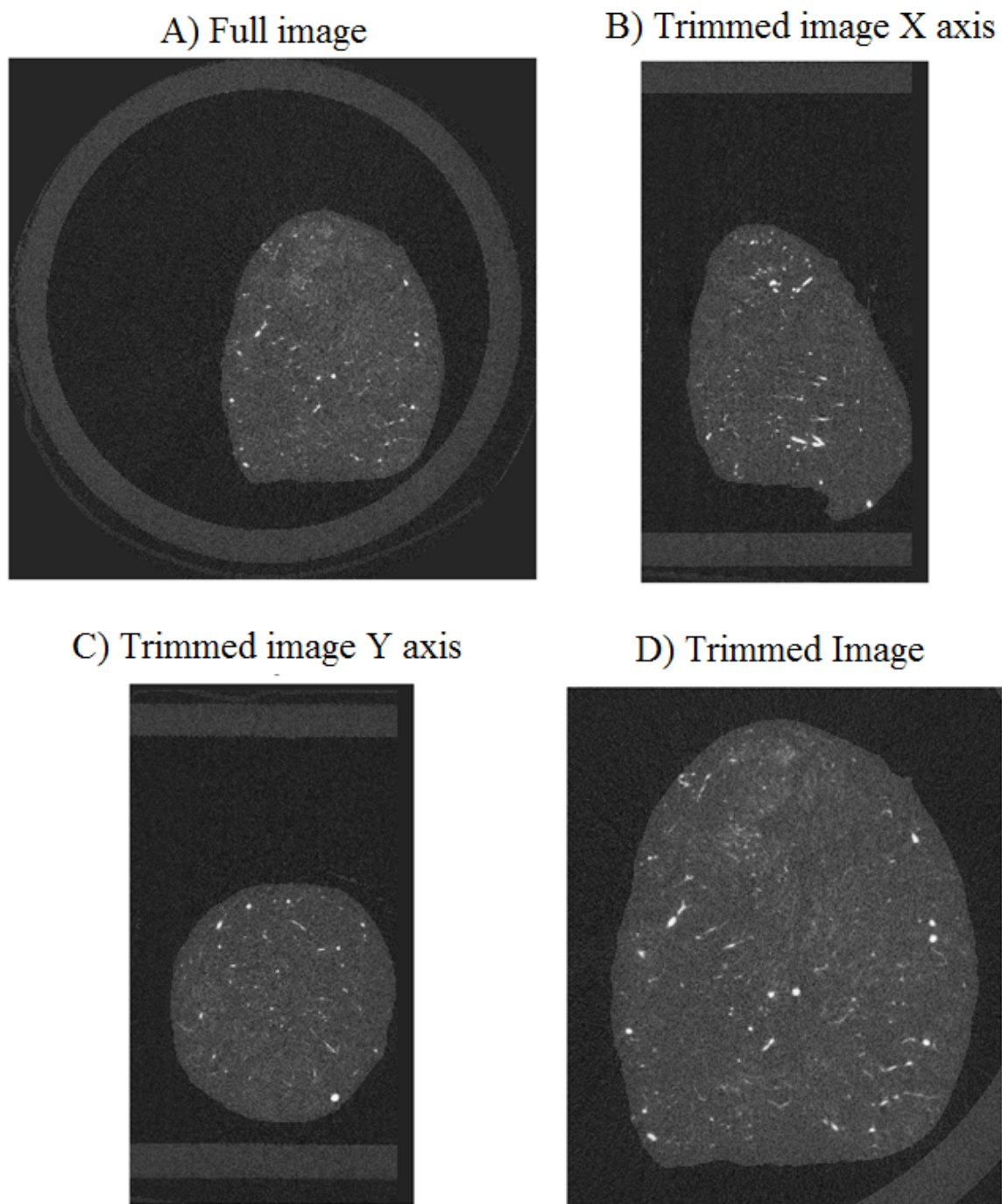


Figure 4.2: Slices and trimmed slice

three things must be provided: 1) a path to the folder containing the image slices, 2) the path to a single image slice, and 3) the total number of slices. The pixel array is extracted slice-by-slice and added to a three dimensional array stacking the slices on top of one another. This does not cause any image skew because the slice proportions are the same, pixel width and height is 6 microns while slice thickness is also 6 microns. The result is a square voxel that can be stacked without distortion. If the slice thickness was not the same as the pixel width and height other steps would be required to correct this distortion before continuing.

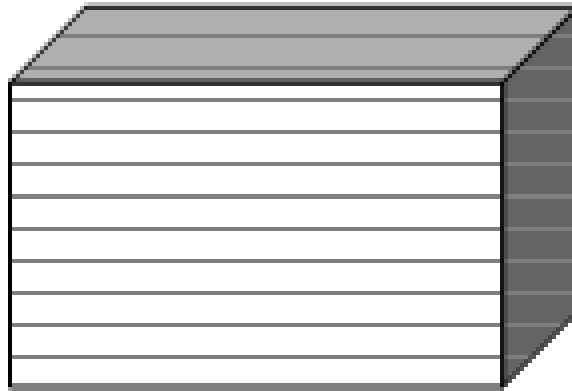


Figure 4.3: Slice Stacking

4.2 Pre-processing

Approximately 3/4 of the areas on the mouse heart images in the scanned set can be removed because they mainly contain background which is considered as noise. This can reduce the memory needed to upload an image set to about 2GB. In addition, this reduces the complexity of the algorithms for reconstructing the 3d volume (cube) containing the

blood vessels and prevents the noise from being used in the future operations as well. The raw image dimensions are 1968x1992x1084 and after trimming the background (noise) the new image size is reduced to 1000x800x1084.

After assembling the cube the next step is to apply a threshold to the image to set the background to a unified standard value. This step allows the background to be separated from the cardiovascular tissue of interest. The background pixel intensity of the background tissue is much lower than that of the cardiovascular tissue allowing for an easy distinction. This step also contributes to increasing the differential between the lower density vessels and the cardiovascular muscle tissue.

4.3 Filtering

After loading only the region of interest, a portion of the vessels can be visualized as illustrated on Figure 4.4. Without the previous steps, it is not feasible to display the image due to the large rendering time and artifacts that exist in the background. The visualization of the full image with these minimal steps taken still can take over 15 hours and is not feasible on a large scale. This visualization provides a means to manually identify abnormalities in an image while also providing feedback on the results of any image processing done. The first applied filter is a straightforward thresholding filter. Pixel intensity values can range from -32,000 to +32,000, but in the case of mouse heart images used in this thesis, pixels are usually between 0 and 17,000 intensity. A band pass threshold can be applied to identify all pixels that are not within a threshold value

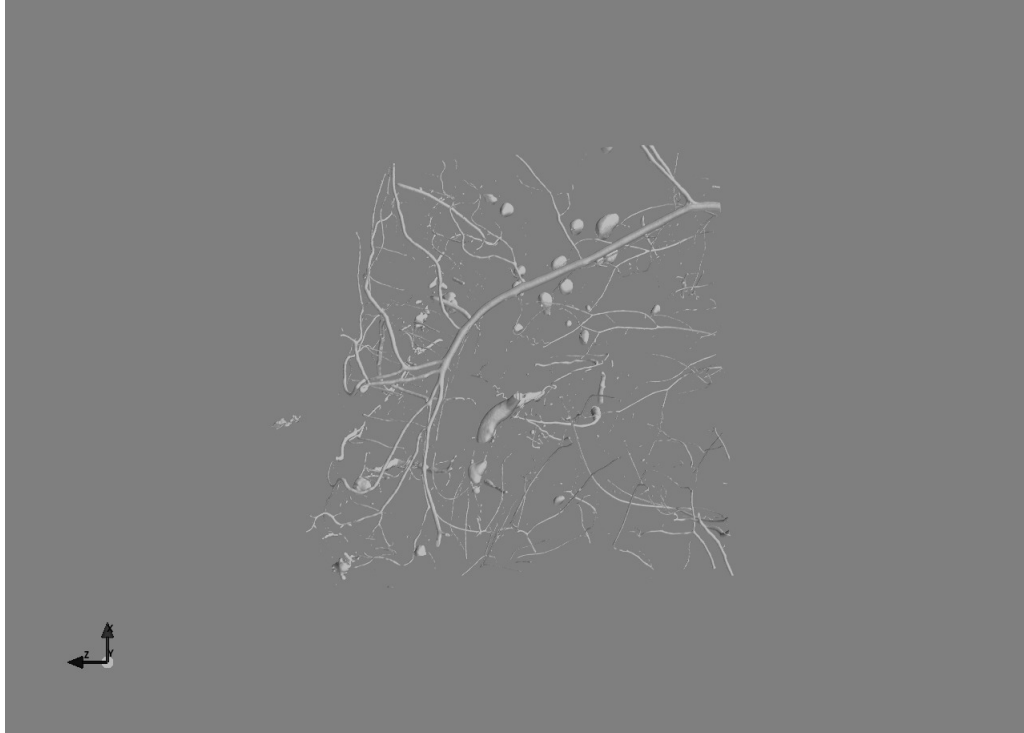


Figure 4.4: Heart Visualization with no processing

of 12,000 to 13,000 and set their values to 0. This is illustrated in 4.5. Changing the threshold value can increase or decrease the size of vessels appearing in the resulting image because larger vessels have a higher intensity than smaller ones. Application of this threshold filter reduces image noise through the removal of fragments not intensive enough to be a vessel but also not like the rest of the cardiac tissue. These fragments could include small air pockets or denser tissues than the majority of surrounding tissue.

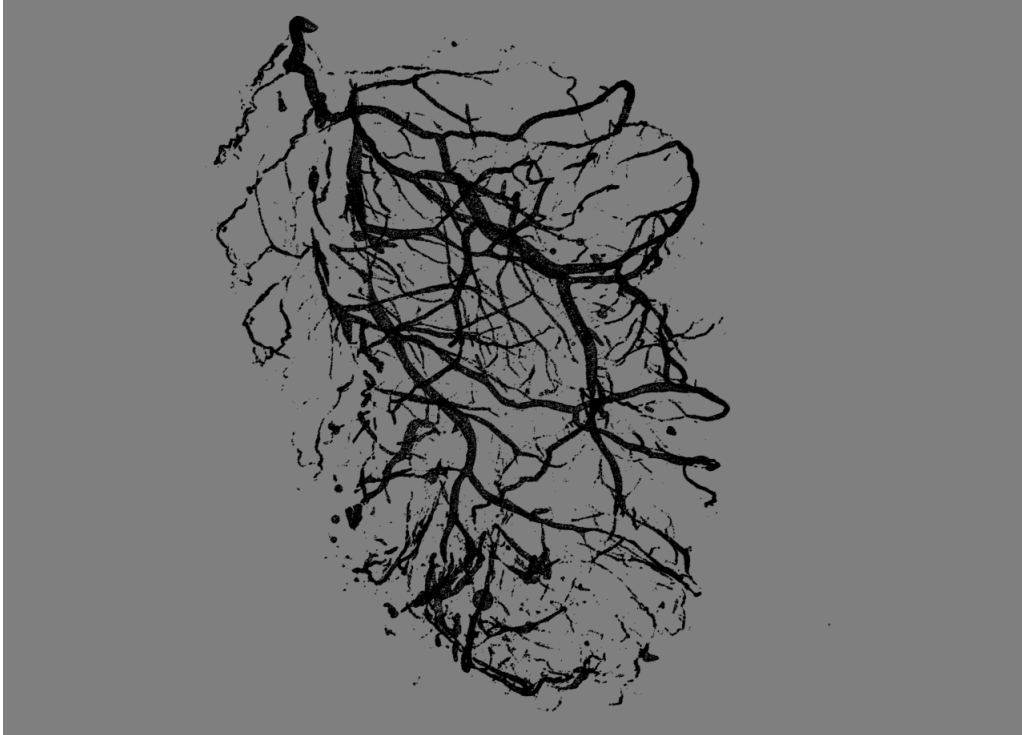


Figure 4.5: Heart Visualization with a Threshold

4.4 Regional Filtering

After pre-processing is complete, the next step is to apply a regional filtering strategy to the image. The difference in vessel pixel intensity varies based on vessel size and location. Due to this variation, a single threshold or filtering plan for the entire image is unfeasible. However, vessels continue to have a higher pixel intensity than the surrounding tissue. Therefore the image is then processed in smaller sub-regions. The number of sub-regions can be changed, but we have used 15 for a full slice. This results in regions of approximately 75 by 60 pixels. Through testing this size range allows for the identification of small vessels while not adding artifacts and vessel structures. The larger number of region caused too many false positives for vessels with the small region sizes. Additionally,

large regions resulted in fewer vessels being identified including the collateral ones which are the primary interest.

While the sub-region size is variable, in practice it is kept consistent for all sub-regions of an image. Testing was done to change the sub-regions dynamically based on average pixel intensity but this resulted in worse results than holding a constant number of regions for each slice of an image. The threshold is calculated dynamically for the local sub-region. The threshold is calculated based on the keeping the top 3 percent of the pixels. The threshold is repeatedly applied until the standard deviation is 1000 less than the average. This serves to remove noise and isolate the vessels. The repetition is used to ensure the less intense pixels are included as those contain some of the smallest vessels. After thresholding, the resulting pixels are used as a mask on the original image. This image is then filtered. First a Sobel edge detection filter is used for all 3 dimensions and is then averaged for each pixel. Finally, a Gaussian filter is applied to smooth the image and reduce any noise as well as remove any artifacts. The resulting image is then visualized in three dimensions based on a visualization toolkit (VTK) routine.

4.5 Region Growing

After the regional thresholding region growing can be attempted. Region growing requires neighboring pixels to be connected and similar. However due to thresholding, image resolution, and potential image discontinuities, this is not always the case. Figure 4.6 highlights a significant example of this. The large vessels occur close to the artery

which causes region growing to end prematurely. These discontinuities present a technical challenge that can be solved but requires extensive work to do so. Removing the discontinuities requires connecting the two vessels or restarting the algorithm on the other side of the discontinuity. Both methods require selecting the other side of the discontinuity. Automatically selecting the correct vessel next to the discontinuity is tricky and is instead currently done manually. While restarting the region growing algorithm would work, it still does not solve the issue of the discontinuity itself. That is considered in the next section using interpolation.

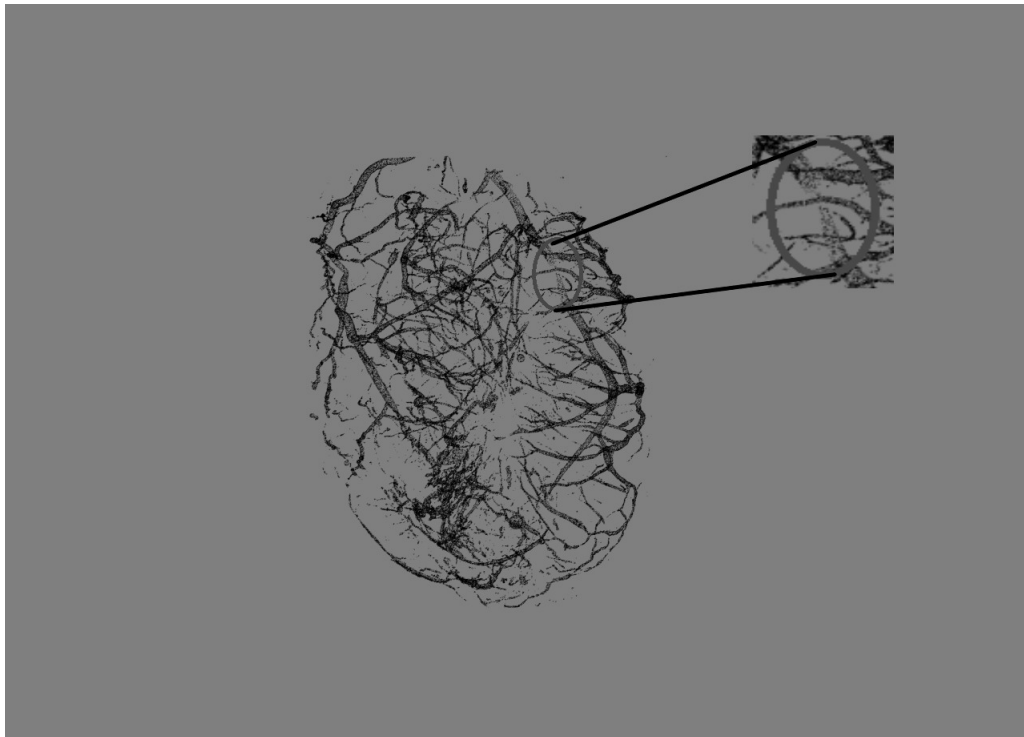


Figure 4.6: Heart Visualization with a Discontinuity

4.6 Interpolation

Interpolation is utilized in this thesis to address the discontinuities that occurred on image frames where visually it was expected to connect 2 vessels. The goal is to provide points of the vessel provided by the existing image frames and use spline interpolation to return a mathematical formula modeling the shape of the vessel. This would allow us to use the mathematical model to connect the two sides of the same vessel on two consecutive frames to establish continuity of the vessel. However the main drawback to a larger scale application of this process is the need to manually select points. Another issue is that a vessel is usually more than 1 voxel in size and interpolation works best with a single point per slice. Therefore each slice must have the vessel reduced to a single point. Linear interpolation works well in most cases, however spline interpolation was chosen because it allows for a more refined interpolation in the cases where the vessel is perfectly straight, however this does require more than two points for an accurate result.

Interpolation was implemented using 4 main steps:

1. Point Acquisition
2. Pattern Search for subsequent slices
3. B-Spline interpolation
4. Prediction for missing slice result

Step 1 consists of point acquisition. This is done manually. A user must provide the point from which interpolation must start. Additional points before this can be included to assist in capturing a more complex curve but are not required. After the

initial point is provided, the algorithm will begin step 2, searching for the most likely point on the next slice. This is achieved by conducting a search in an outward spiral pattern on the next slice from the point provided in step one. Figure 4.7 shows an example of the pattern.

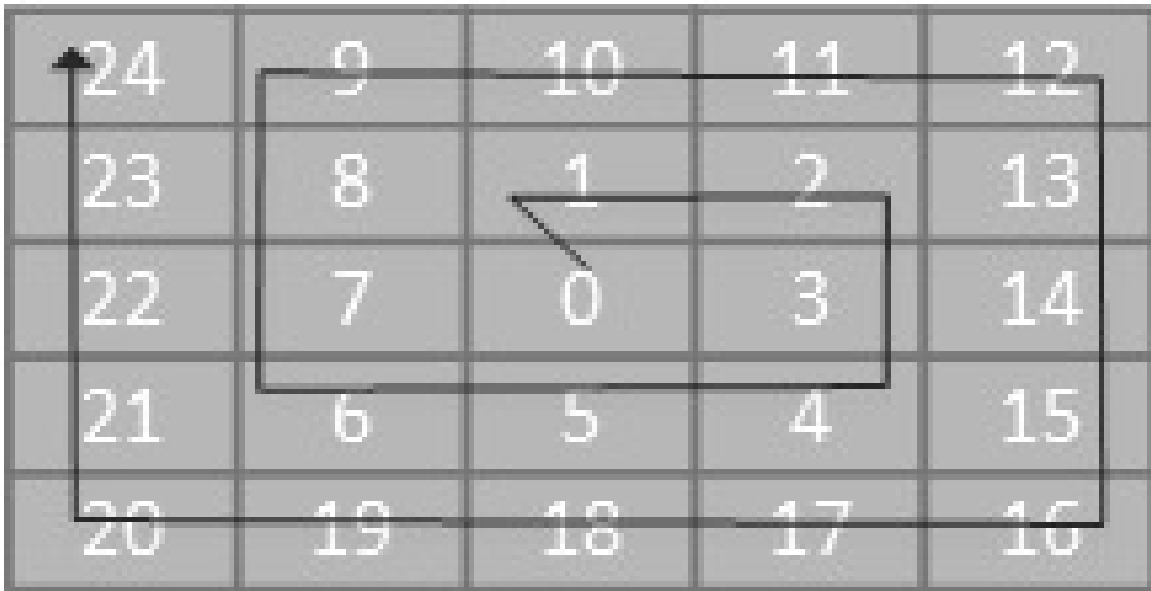


Figure 4.7: Search pattern for subsequent slices

This works because the vessels do not drastically change orientation across each slice. The number of slices it searches can be set to allow for a more complex curve to be calculated if needed. After an adequate number of slices are captured, the points are then used to calculate a B-Spline interpolation equation. Then, for the slices that contain the discontinuities, the equation is used to predict where the vessel would be. This only returns a point, however; thus the radius of the vessel has to still be estimated by calculating the radius of the vessels at surrounding points. Currently, only the point

is used to fix the discontinuity and the radius is not calculated. Without the estimated radius it is not possible to replace more than the initial point calculated within the image.

After the search is completed. The points that are found are used to do spline interpolation to create a equation that represents the suspected vessel path. The expected curve is not drastic and has few inflection points so a low order interpolation is sufficient. An order of 5 is used to ensure adequate flexibility in the polynomial equation. Finally, after the equation is generated, slices where the data is missing or unclear can be used in conjunction with the equation to find the expected position of the vessel.

4.7 3D Display

The display method used to render the image is a 3D contour map. A contour map provides proper visualization because the vessels have a much higher intensity than the surrounding tissue. A VTK (The Visualization Toolkit) object is created from the image data array. Four contours are plotted to render the vessels. With less than four contours, small fine vessels like collateral vessels are lost and not clearly displayed. In addition, using more than four contours causes artifacts, non- vessel fragments, and other non-vessel structures to appear. The number of contours needed for optimal visualization is dependent on the processing and image analysis done beforehand. However, for the methods previously described the the best rendering results from using only four contours.

Chapter 5

Results

Identifying and segmenting blood vessels in the images of a mouse heart is a challenging problem that can only be achieved using a combination of several methods. This research utilizes several of these methods in varying combinations to address different issues and problems that arise during segmentation. This chapter presents the results after the full workflow is complete and for some intermediate steps.

5.1 Preprocessing

The first step after loading one of the images from the DICOM files is to trim it by removing most of the background that doesn't contain any cardiovascular information. This step speeds up the processing time and prevents the background from overly influencing the calculations. An example is shown in Figure 5.1 achieved by setting a threshold of 500.

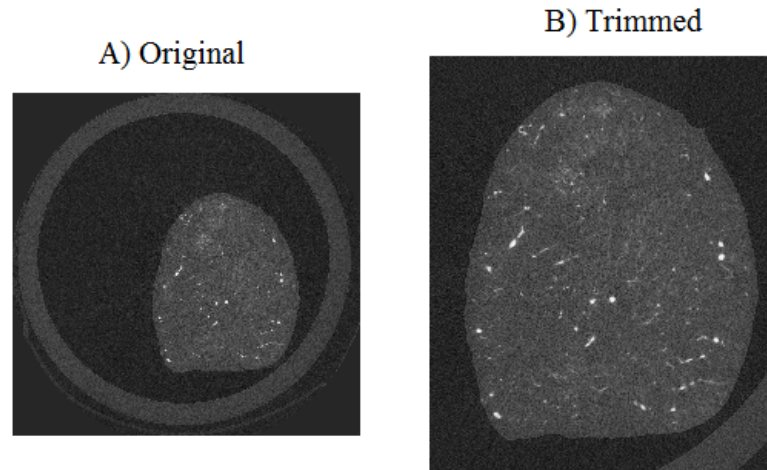


Figure 5.1: Example of background removal

After trimming the image, the background pixels are separated using a threshold value and all are set to 0 which represents the color black. In the case of this research, the background is considered the area surrounding the diagnostic part of the image with pixel intensity values of lower than 500. Figure 5.1 illustrates the increased contrast resulting after the completion of of the background removal. Setting the background to a set color such as black helps eliminate a significant number of artifacts. This preparation will improve processing speed in later steps as well.

5.2 Regional Results

After the completion of the preprocessing step, each image slice is divided into 9 subregions which were determined empirically based on the diversity of the intensity values appearing in the image. For example, in one region a vessel could appear as bright as a pixel value of 15,000 while the continuation of the same vessel to the side of the image had a pixel value of 8000. This is partially because the regions away from the center of the image appear with lower intensity values, i.e. are dimmer. The highest value pixel intensities are considered to be vessels in the subregion, thus a region's specific threshold value is applied on each of these regions to separate the vessels. This threshold value is determined by investigating all the pixel values in that region and finding the highest intensity value in that region for a vessel and using that as the base for the threshold value with a small percent tolerance. Figure 5.2 illustrates the results on a single slice. The thresholds for these regions vary between 800 and 15000 across the entire image set. Some slices had thresholds values varying by as much as 4000. These threshold values highlight the variance across the full image and demonstrate a need to treat the images as subregions instead of applying a threshold to the entire image.

Figure 5.2 contains an example of 9 regions from a slice. The squares labeled a through i are the original images. The squares labeled a1 through i1 correspond to these images, respectively, after a regional threshold has been applied. This method results in many more vessels being identified. The highest intensity spots on the original image are shown as white. These white spots also correspond to the white spots remaining after

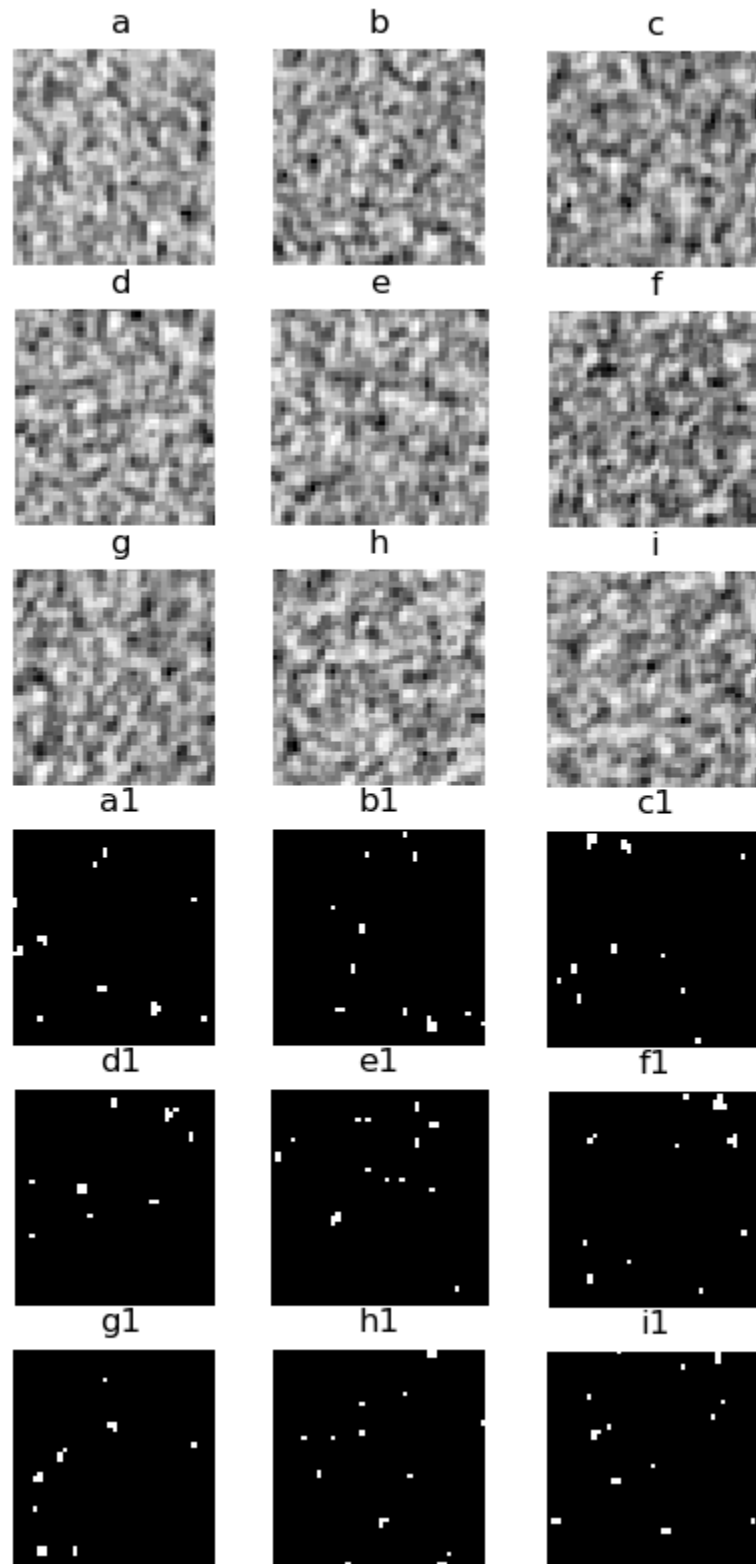


Figure 5.2: Regional Filtering: a-i - original images and a1-i1 - their corresponding results

thresholding. However, some spots that are not as obvious are also identified. The spots that are not as high intensity are the ones that gained by using an adaptive regional threshold instead of a global one.

5.3 Threshold Results

Applying a single threshold value to the entire image is moderately successful in identifying vessels that are clearly bright on the image. However, it fails to identify and extract many small vessels that are not as bright. The global threshold approach was used to separate background from cardiovascular tissue and the threshold value was adjusted based on the image slice to achieve the highest accuracy.

Figure 5.3 shows the results of applying a global threshold to a single slice. The slice is displayed on both dimensions. The global threshold isolates the larger vessels but fails to isolate the smaller ones. Figure 5.2 shows the results of the regional filtering. The regional filter is able to isolate an increased number of vessels because the threshold is adaptive to the region based on the region of the image and can find the smaller vessels at a lower intensity.

5.4 Interpolation Results

Interpolation can be used to predict data points in missing or incomplete slices. In addition, it can be used to correct disconnects in vessels due to thresholding or other processes. These situations are unavoidable in some cases as parts of the vessel may be

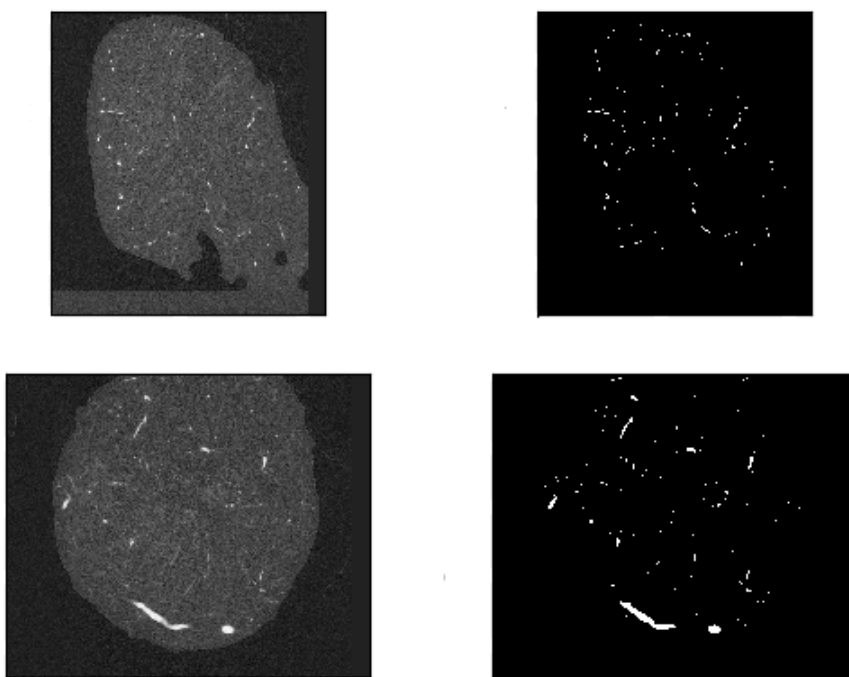


Figure 5.3: Results of Global Threshold

at a lower intensity due to non-standard factors. Some of these factors include denser tissue surrounding the vessel, location of vessel (closer to the center of the heart). Linear interpolation was able to predict and connect sample data, however it worked best for only one missing point. For times when multiple slices were missing, b-spline interpolation was preferred. This method handles the slight curve vessels can exhibit even over a single slice. Part of this process uses the interpolate package in *scipy*.

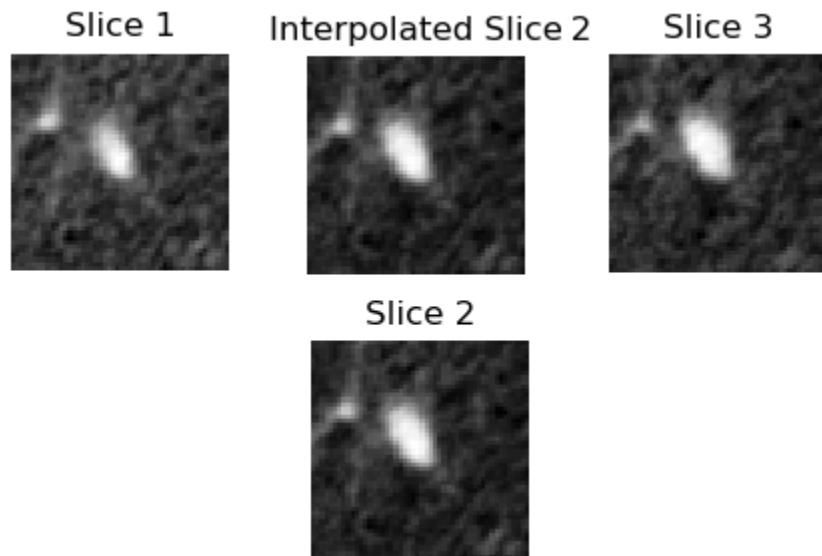


Figure 5.4: Example of Interpolated Missing Slice

For interpolation to work well, the known points must be extracted from the scanned images. A single point per slice is required to achieve the best results. These points were extracted manually because the challenge of reducing vessels to a single center point per slice is out of the scope of this project. Searching for subsequent points given a single starting point was possible but required the vessels to be reduced to a single point. This search method shows promising results but requires that vessels be reduced

to a single point for every interpolation. Manually finding a point representing every vessel per slice is extremely time consuming and not feasible for large scale images. For this reason, using interpolation to connect vessel fragments was not used to analyze the entire image despite its promising results. Another problem with interpolation is that the slices are taken every 6 microns and often the small blood vessels have a diameter in a lower range. To solve this issue, slices should be obtained at a lower spacing.

Figure 5.4 is an example of how interpolation can be used to substitute for a corrupted, missing, or inaccurate slice or region. Some information may be lost if the interpolation model doesn't perfectly match the trend of the vessels from one slice to another, but interpolation allows for a close estimate to be created for the missing value. Linear interpolation is used in this case as only 2 data points from two subsequent slices were used to reconstruct the missing pixel. Figure 5.5 illustrates how interpolation can be used in our research to generate a new slice between two known slices. Note that for the illustration purpose here, we have taken a known slice (Slice 2) out from between two other known slices (Slice 1 and Slice 3) and reconstructed Slice 2 using this technique to come up with Interpolated Slice. As it can be seen the Interpolated Slice and Slice 2 are almost identical. However, the interpolated slice contains pixels 10% less than the original. This results in a darker figure and less contrast between vessels and background tissue. Figure 5.5 illustrates the reconstruction of a new slice based on the two known slices in the set.

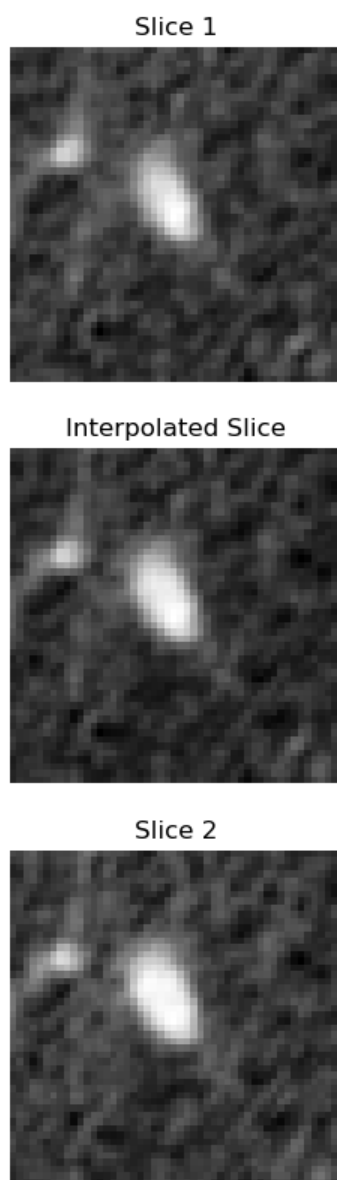


Figure 5.5: Example of Interpolation between Slices

5.5 Filter Results

After the images were trimmed and thresholded filters were applied to the data, a Sobel and Gaussian filter were both applied on the resulting images. This combination is used to help find and sharpen the vessel edges but also to connect and smooth them. Originally this was used to help the region growing process and bridge the gaps across fragmented vessels. This method was not suitable for all gaps and did not allow region growing to complete. Thus, this step was used to refine the vessels and fill in gaps before display. By using this combination of filters, the edges were refined. A mask was then created for the original image and was used to further isolate the vessels and remove extra fragments. This allowed for 3D rendering to be completed in a reasonable time since only areas identified by these filters were displayed.

Figure 5.7 contains more complete vessels and smaller vessels than Figure 5.6 or Figure 4.5. Figure 5.7 shows the results of the refinement after the regional threshold is applied to Figure 5.6. Specifically in the smaller, thinner vessels less discontinuities are found and these vessels show a better connection to the other vessels.

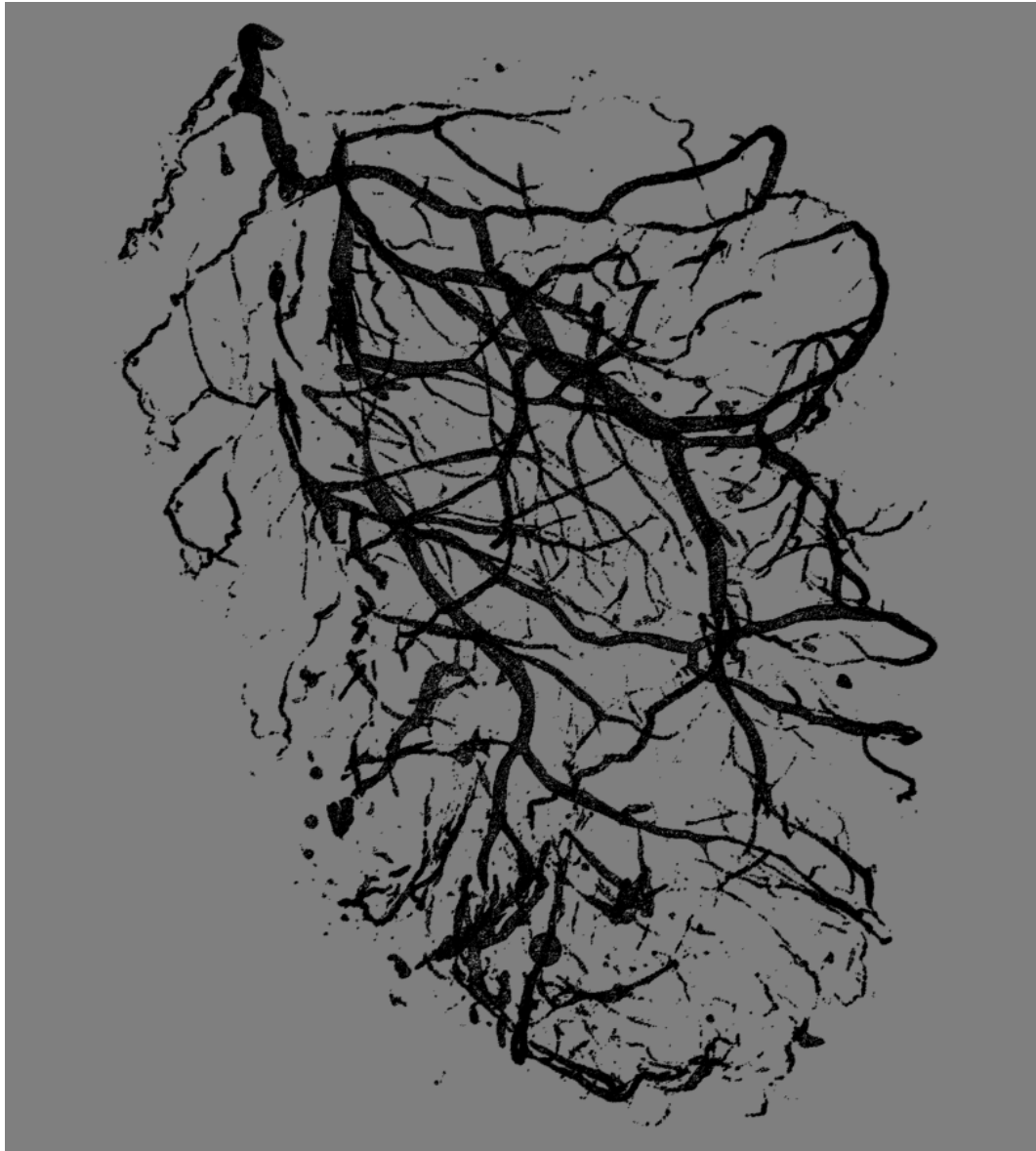


Figure 5.6: Display before filtering

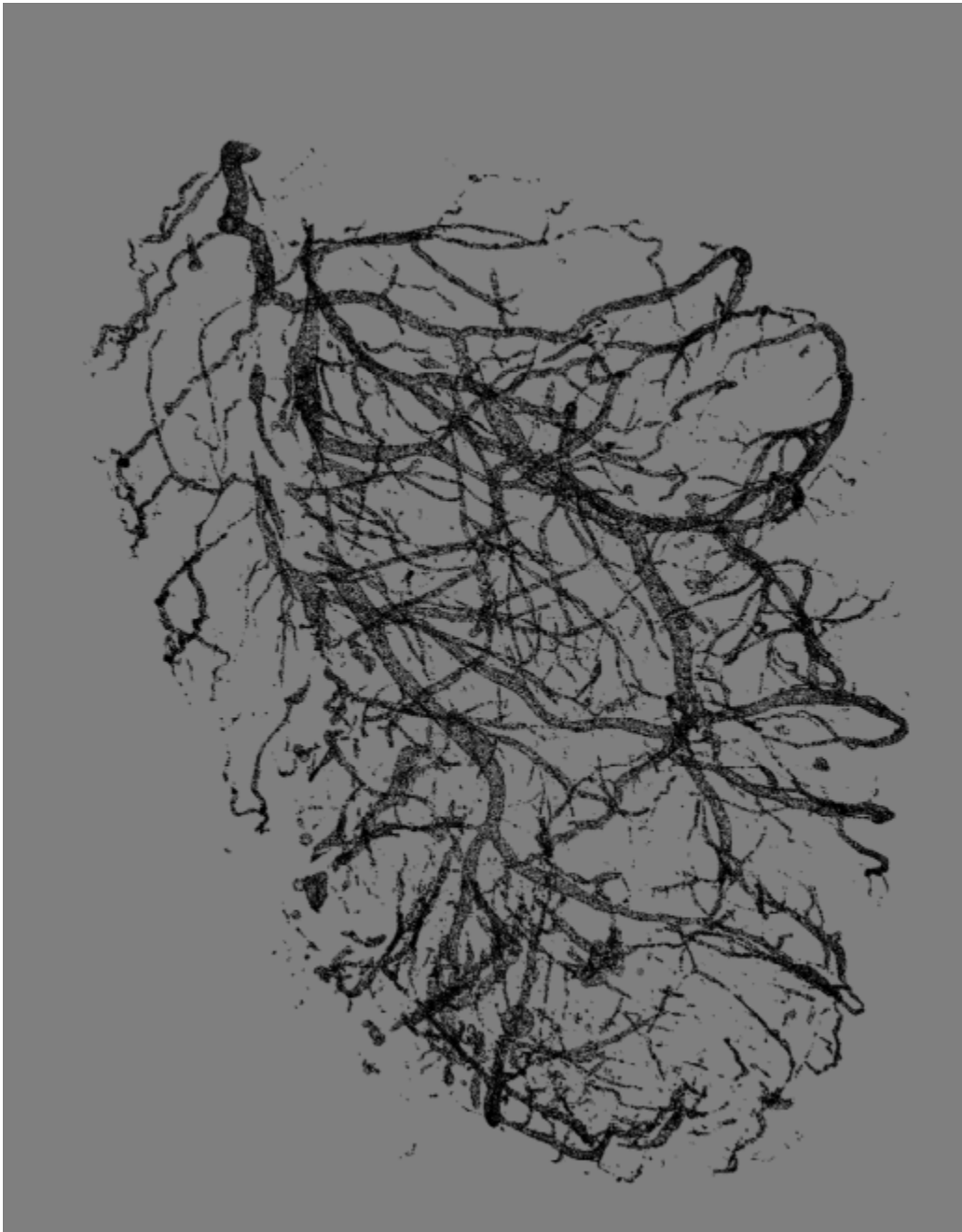


Figure 5.7: Display after filtering

Chapter 6

Conclusion and Future Work

6.1 Preprocessing

The preprocessing applied on the images used in this research was an important step for reducing computation time and allowing for accurate threshold values to be acquired. Figure 5.1 illustrates the reduction in size. In addition, Figure 5.1 shows the increase in contrast achieved by isolating the background values. However, because the background shares some pixel values within the region of interest(ROI), it is important to note that the ROI is being slightly modified. This modification is not an issue as no vessel appears with a pixel value close to the background.

6.2 Regional Filter

The regional filter is an appropriate way to filter the image slices. While a global threshold was initially proposed, it was not feasible. Thus, the regional filter that could adapt to

each subregion provided increased feature distinction. Therefore it was superior to a single global threshold to identify vessels. Figure 5.2 shows the results on one of the images slices. Making the threshold value too high resulted in only the largest vessels with the highest intensity values to remain, while too low of a value resulted in non-vessel artifacts to appear as vessels. With this in mind, for the images used in our research, a threshold value at 99th percentile of the highest intensity value observed within a region was found to be most suitable through an empirical trial and produced reasonable results in the final display.

6.3 Sobel and Gaussian Filtering

After thresholding was preformed, the image was not ready to be displayed. Applying Sobel and Gaussian filters to the image, resulted in smoothing the edges and refining the image. The creation of a mask with smoothed edges also allows for identifying more complete and connected vessels. While this step is not necessary for accuracy it does facilitate a more efficient display of the image. Without this step, the image would take more memory and time to display.

6.4 Interpolation and Region Growing

Interpolation and region growing were both proven useful and showed great potential for assisting in segmentation of the vessels in the mouse heart images. However, these methods require additional investigation to produce reliable and accurate results for

this project. In example cases with manual adjustments, it was illustrated that they were useful in refining vessels and isolating vessel segments. Due to the additional work required for interpolation to extract the points by reducing a vessel to a single pixel per slice, it is not feasible to successfully use it in a fully automated process, but it shows a significant potential for future work. Region growing works best when the vessels are fully connected and no fragments exist. This state is not likely so the disconnects and fragments must be addressed to acquire a full representation of the vessel structure. Solving these fragments and disconnects problem requires utilizing interpolation, different preprocessing steps, or other filtering methods.

6.5 Future Work

The research uncovered the many challenges related to segmentation of fine blood vessels in the heart of a mouse. A significant amount of work needs to be done to achieve reconstruction of these vessels to create an accurate 3D model. For one, a tool could be developed to make threshold determination and seed selection faster and easily applicable to different image sets. Currently, this step must be done manually and is not precise. In addition, the tool could be designed in such a way that it can be applied to blood vessels of various diameters. Another work that must be expanded upon is allowing for the tracking of different coronary vessels.

This thesis provides a means to study coronary collateral vessels through medical imaging. Future research into the formation or loss of collateral vessels over time would

be made easier with the results from this work. Researchers could investigate stimuli affecting the angiogenesis for collateral vessels, including drugs and exercise.

Bibliography

- [1] CDC. Underlying cause of death 1999-2013 on cdc wonder online database, released 2015. data are from the multiple cause of death files, 1999-2013, as compiled from data provided by the 57 vital statistics jurisdictions through the vital statistics cooperative program. accessed may 2, 2018.
- [2] W. M. Chilian, H. J. Mass, S. E. Williams, S. M. Layne, E. E. Smith, and K. W. Scheel. Microvascular occlusions promote coronary collateral growth. *American Journal of Physiology-Heart and Circulatory Physiology*, 258(4):H1103–H1111, 1990. PMID: 2330998.
- [3] X. Jiang and D. Mojon. Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25:131–137, 2003.
- [4] J. Jin, L. Yang, X. Zhang, and M. Ding. Vascular tree segmentation in medical images using hessian-based multiscale filtering and level set method. *Computational and mathematical methods in medicine*, 2013:502013, 11 2013.
- [5] S. Jyothi and K. Bhargavi. A survey on threshold based segmentation technique in image processing. 3, 11 2014.
- [6] R. Malladi, J. A. Sethian, and B. Vemuri. Shape modeling with front propagation: A level set approach. 17:158 – 175, 03 1995.
- [7] P. C. Marks, M. Preda, T. J. Henderson, L. Liaw, V. Lindner, R. E. Friesel, and I. M. Pinz. Interactive 3d analysis of blood vessel trees and collateral vessel volumes in magnetic resonance angiograms in the mouse ischemic hindlimb model. In *The open medical imaging journal*, 2013.
- [8] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12 – 49, 1988.
- [9] M. Paulus, S. Gleason, S. Kennel, P. Hunsicker, and D. Johnson. High resolution x-ray computed tomography: An emerging tool for small animal cancer research. *Neoplasia (New York, N.Y.)*, 2:62–70, 04 2000.

- [10] R. Rudyanto, S. Kerkstra, E. van Rikxoort, C. Fetita, P.-Y. Brillet, C. Lefevre, W. Xue, X. Zhu, J. Liang, I. Oksuz, K. Unay, R. Estepar, J. Ross, G. Washko, J.-C. Prieto, M. Hernandez Hoyos, M. Orkisz, H. Meine, and B. Ginneken. Comparing algorithms for automated vessel segmentation in computed tomography scans of the lung: The vessel12 study. *Medical Image Analysis*, 18, 07 2014.
- [11] M. Schaap, T. van Walsum, L. Neefjes, C. Metz, E. Capuano, M. de Bruijne, and W. Niessen. Robust shape regression for supervised vessel segmentation and its application to coronary segmentation in cta. *IEEE Transactions on Medical Imaging*, 30(11):1974–1986, Nov 2011.
- [12] C. Seiler, M. Stoller, B. Pitt, and P. Meier. The human coronary collateral circulation: Development and clinical importance. *European heart journal*, 34, 06 2013.
- [13] E. Tatli, A. Altun, M. Büyüklü, and A. Barotçu. Coronary collateral vessel development after acute myocardial infarction. *Experimental and clinical cardiology*, 12:97–9, 07 2007.
- [14] Y. Yang. Image segmentation and shape analysis of blood vessels with applications to coronary atherosclerosis. 03 2019.
- [15] N. M. Zaitoun and M. J. Aqel. Survey on image segmentation techniques. *Procedia Computer Science*, 65:797 – 806, 2015. International Conference on Communications, management, and Information technology (ICCMIT’2015).

Vita

Erik John Cole was born in 1992 in Iowa City, Iowa, to Kelly and Karalee Cole. He graduated from Solon Community High School in Iowa in May 2010. The following fall, he entered the Bio-Medical Engineering program in the College of Engineering at The University of Iowa. He graduated in 2015. In 2016, he enrolled at Appalachian State University to pursue a Master of Science degree in Computer Science.