

January 2014

Extracting Vessel Structure From 3D Image Data

Yuchen Zhong

The University of Western Ontario

Supervisor

Yuri Boykov


The University of Western Ontario

Graduate Program in Computer Science

A thesis submitted in partial fulfillment of the requirements for the degree in Master of Science

© Yuchen Zhong 2014

Follow this and additional works at: <https://ir.lib.uwo.ca/etd>

 Part of the [Cardiovascular System Commons](#), [Geometry and Topology Commons](#), [Graphics and Human Computer Interfaces Commons](#), and the [Other Computer Sciences Commons](#)

Recommended Citation

Zhong, Yuchen, "Extracting Vessel Structure From 3D Image Data" (2014). *Electronic Thesis and Dissertation Repository*. 1883.
<https://ir.lib.uwo.ca/etd/1883>

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact tadam@uwo.ca.

EXTRACTING VESSEL STRUCTURE FROM 3D CARDIAC IMAGE
(Thesis format: Monograph)

by

Yuchen Zhong

Graduate Program in Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

© Yuchen Zhong 2014

Abstract

This thesis is focused on extracting the structure of vessels from 3D cardiac images. In many biomedical applications it is important to segment the vessels preserving their anatomically-correct topological structure. That is, the final result should form a tree. There are many technical challenges when solving this image analysis problem: noise, outliers, partial volume. In particular, standard segmentation methods are known to have problems with extracting thin structures and with enforcing topological constraints. All these issues explain why vessel segmentation remains an unsolved problem despite years of research.

Our new efforts combine recent advances in optimization-based methods for image analysis with the state-of-the-art vessel filtering techniques. We apply multiple vessel enhancement filters to the raw 3D data in order to reduce the rings artifacts as well as the noise. After that, we tested two different methods for extracting the structure of vessels centrelines. First, we use data thinning technique inspired by Canny edge detector. Second, we apply recent optimization-based line fitting algorithm to represent the structure of the centrelines as a piecewise smooth collection of line intervals. Finally, we enforce a tree structure using a *minimum spanning tree* algorithm.

Keywords: Vesselness Measure, Hessian Matrix, Gaussian Derivatives, Harris Corner Detector, Eigenvalue Decomposition, Canny Edge Detector, Model Fitting, Rings Reduction, Noise Reduction, 3D Volume Visualization, Minimum Spanning Tree

Acknowledgement

First of all, I want to express my sincere gratefulness to my Professor Yuri Boykov, without whom I will never be able to write this thesis. I felt extremely lucky to have him as my supervisor. He is a great mentor, as well as a great researcher and speaker. He has given me a lot of guidance during the one year and a half. My public presentation for the final thesis would have been much worse without all those so important feedback from. His attitude toward teaching, research and life has a great impact on me.

I am also very grateful to Professor Olga Veksler who is always so thoughtful and kind. Even though officially she is not my supervisor, but she is always there to help. Whenever we have results that are not so satisfying, she always uses her smile to make feel better and her words to encourage us to achieve more.

And also, Lena, Claudia and Eno, who have been acting as our temporary supervisors while Yuri and Olga were away last summer. Thanks for all the feedback, comments and suggestions during the summer.

The original data of this thesis is provided by Maria Drangova. Thanks for her data and for sharing all her knowledge about blood vessels. Looking forward to our future cooperation.

I never understand Fourier transform during my undergraduates. But I have a much deeper understanding after taking John Barron's course *Image Processing*. And also I want to thank him for his extremely detailed and constructive comments for my thesis.

Special thanks also goes to Janice Wiersma and Professor Laura Reid for nominating me for the TA awards. That means a lot to me. Being a TA is one of the most interesting things to me. It was also a great pleasure to know AbdulWahab Kabbani, who had been working with me during our consulting hours together. He is also a great TA and a very good speaker. It made my life a lot easier to team up with him for consulting. I enjoyed talking to him in the lab when no student came. Sometimes, when assignment is near due, he would stay a little bit longer to help me with my section and I really appreciate that.

I am very happy to know Meng, Xuefeng and Liqun, who are the Chinese students in our lab. Thanks for hanging out with me for so many time. It is because of them that I don't miss my hometown that much. I like listening to Meng talking about his new ideas excitingly. And I am also very grateful to Liqun for reading my thesis and providing a lot feedback. Thanks to Xuefeng for working on this not-so-easy project with me. And it is also very lucky to know Enxin Wu, who is a PhD in math. He spend a lot of time helping me understand some of the mathematical concepts such as Hessian, Conic and Quandric, integration of ellipse and etc. I cannot forget the time when Junwei was still here. I enjoy biking around the river and playing soccer with him during the whole summer. After he left, it is hard for me to find another friend

who is so passionate about sport and can play with me all the time. Wish you all the best at Toronto. Today is Chinese New Year. Happy new year to all my Chinese friends!

I am very happy that I meet Igor Milevskiy. It is very nice to have a ‘labmate’ like him. Thanks for letting me know so much about steaks. I also enjoy his stories about Korea, Hongkong, Beijing, Japan, and, of course, Russia. After all, I spend both of the Thanksgiving Dinner with him during my two years here. Thanks for buying me Lunch and Coffee when I am sometimes out of cash. Thanks for sharing his knowledge about the algorithms for computing *minimum spanning tree*.

Thanks to my lovely Shuaishuai Yuan, only who would spend that much effort trying to read and understand my thesis starting with the first draft version. Love her as always.

Finally to my great parents. I have been away from home for too long. Miss them and thanks very much for all their support and understanding during my whole life.

Contents

Abstract	ii
Acknowledgement	iii
List of Figures	viii
List of Tables	x
List of Appendices	xi
1 Introduction	1
1.1 Problem Overview	1
1.1.1 Input and Ideal Result	2
1.1.2 Technical Problems	5
(A) Filtering (Preprocessing)	5
(B) Structure Extraction	5
(C) Topological Constraints	8
1.1.3 Pipeline of The Algorithms	9
1.2 Related Work	10
1.2.1 Rings Filtering	10
1.2.2 Vesselness Filtering	12
1.2.3 Graph Cut Segmentation	15
1.2.4 Centreline-Based Methods	16
1.3 Contributions	18
1.4 Outline of the Thesis	18
2 Vesselness Measure	19
2.1 Overview	19
2.2 Harris Corner Detector	20
2.2.1 Derivation of The Matrix for Harris Detector	20

2.2.2	Eigenvalues and Eigenvectors of Harris Detector	21
2.2.3	Visualization of Eigenvalues and Eigenvectors	22
2.2.4	Combination of Eigenvalues	23
2.3	Conceptual Vessels in Different Dimensions	24
2.3.1	3D Vessels as Tubes	24
2.3.2	2D Vessels as Rectangles and Balls	24
2.3.3	1D Vessels as Box Functions	25
2.4	Hessian Matrix and Gaussian Derivatives	25
2.4.1	Second Derivative of Gaussian in 1D	25
2.4.2	Hessian for Rectangles in 2D	31
2.4.3	Hessian for Balls in 2D	35
2.4.4	Hessian for Vessels in 3D	39
2.4.5	Hessian Matrix in General	39
2.5	Combination of Eigenvalues	41
2.5.1	3D Vesselness Measure	41
2.5.2	2D Ballness Measure	42
2.6	Comparison Between Scale	44
2.7	Result	45
3	Centreline Extraction	48
3.1	Overview	48
3.2	Vessel Thinning	49
3.2.1	Non-maximum Suppression	49
3.2.2	Hysteresis Thresholding	50
3.3	Model Fitting	51
3.3.1	Line Interval Fitting with PEARL framework	52
	Propose	52
	Expand	52
	Re-estimate Labels	53
3.3.2	Ball Fitting	53
	Propose	54
	Expand	54
	Re-estimation	54
3.4	Result	56
4	Minimum Spanning Tree	60
4.1	Prim's Algorithm and Kruskal's Algorithm	60

4.2	Minimum Spanning Tree for Points	61
4.3	Minimum Spanning Tree for Lines	63
4.4	Results	63
5	Conclusion	66
5.1	Pipeline of The Algorithm	66
5.2	Future Work	66
	Bibliography	68
A	Visualization of 3D Data	73
A.1	Maximum Intensity Projection	73
A.2	Arbitrary Cross Section	75
B	Rings Reduction	77
C	Eigenvalues of a Symmetric Matrix	80
D	Distance in 3D Space	82
D.1	Distance Between Point to Line in 3D	82
D.2	Distance Between Two Lines in 3D	83
D.3	Distance Between Two Line Intervals in 3D	85
	Curriculum Vitae	86

List of Figures

1.1	Four Slices of the Original Data	3
1.2	Original Data With Maximum Intensity Projection	4
1.3	Original Data and Ideal Result	4
1.4	Rings Artifacts	6
1.5	Rings Reduction	7
1.6	Vesselness Filter	7
1.7	Image Segmentation with Thresholding	7
1.8	Discrete Centrelines	8
1.9	Segmentation Without Connectivity [43]	9
1.10	Centre Line With Data Points	9
1.11	Centre Line With Line Intervals	10
1.12	Pipeline of The Algorithms	11
1.13	Rings Reduction Method	13
1.14	Vesselness Measure [19]	14
1.15	Over-smoothing of Thin Structure With Graph Cuts [25]	15
1.16	Connectivity Constraint in Graph Cut [43]	16
1.17	4D Path Result [28]	17
2.1	Visualized Eigenvalues with Ellipsoid	23
2.2	Conceptual Vessels in 2D	25
2.3	1D Box Function	26
2.4	Gaussian (Blue), 1 st and 2 nd derivative of Gaussian (Green, Red)	27
2.5	Find the best match of 2 nd derivative of gaussian $\frac{\partial^2 G}{\partial x^2}$ for box function	29
2.6	Convolution of 1D Box Function With 2 nd Derivative of Gaussian	30
2.7	2D Vessels	32
2.8	Eigenvalues of The Hessian Matrix for 2D Vessels	34
2.9	2D Balls	35
2.10	First and Second Derivative of Gaussian in 2D	36
2.11	Eigenvalues of The Hessian Matrix for 2D Balls	37

2.12	Different Ways of Combination of Eigenvalues of Balls	38
2.13	Ballness	43
2.14	Vesselness With Different Sigmas	45
2.15	Comparing Original Data and Vesselness	46
2.16	Comparing Original Data and Vesselness (Cross Sections)	47
3.1	Different Situations When Comparing Neighbours With Non-maximum Sup- pression in 3D	51
3.2	Pairwise Interaction Approximating Curvature [34]	53
3.3	Vessel Thinning	57
3.4	Line Interval Fitting	58
3.5	Ball Fitting	59
4.1	Minimum Spanning Tree on Connected Graph	61
4.2	Original Graph	61
4.3	Prim's Algorithm	62
4.4	Kruskal's Algorithm	62
4.5	Minimum Spanning Tree on Discrete 2D Points	64
4.6	Minimum Spanning Tree on 2D Lines	64
4.7	Minimum Spanning Tree on Data Thinning	65
4.8	Minimum Spanning Tree on Line Intervals	65
5.1	Pipeline of The Algorithms	67
A.1	Maximum Intensity Projection	73
A.2	Comparing Between Normal Projection and Maximum Intensity Projection	74
A.3	Cross Section of a Box	75
A.4	Cross Section of The Vessel Volume	76
B.1	Rings Reduction in Polar Coordinates [40]	78
B.2	Comparison Before and After Rings Reduction	79
D.1	Distance From Point to Line in 3D	82
D.2	Distance Between Lines in 3D	83

List of Tables

2.1	Corresponding Shapes for Vessels in 3D, 2D and 1D	24
3.1	Non-maximum Suppression in 3D	50

List of Appendices

Appendix A Visualization of 3D Data	73
Appendix B Rings Reduction	77
Appendix C Eigenvalues of a Symmetric Matrix	80
Appendix D Distance in 3D Space	82

Chapter 1

Introduction

1.1 Problem Overview

This thesis focuses on extracting the topological structure of a 3D cardiac images. It consists of three main parts: (a) image filtering to remove noise; (b) extracting the centreline of vessels; (c) enforcing a tree structure for the vessel with *minimum spanning tree*.

Extracting vessel structure remains a challenging problem because of a number of technical problems. Due to partial volume, the intensity of small vessels become weaker or even completely disappear. Acquisition artifacts, such as rings or random noise, are very common in the data. Special image filtering is required to remove these artifacts from the data while preserving the details of small vessels. Even after these filtering, it is still not easy to extract the image structures which can be either the segmentation or the centreline of the object. Topological constraints are enforced on the image structures in order to remove ambiguities in the result. These technical problems are further discussed in Section 1.1.2.

Standard methods have problems in extracting vessel structures with topological constraints. Graph cuts [6] is a recent optimization-based algorithm for image segmentation. But its over-smoothing problem tend to smooth out the thin structures. Different attempts are made to address this over-smoothing problem (see Section 1.2.3). An alternative approach for extracting vessel structure is by extracting the centreline. This can be achieved by calculating the minimum path between two user input points. This will be further discussed in Section 1.2.4.

It is not easy to validate the algorithms for extracting vessel structures. Therefore different visualization methods are developed so that we are able to see the 3D volume better. Two visualization methods are frequently used in this thesis: (1) *maximum intensity projection*; and (2) visualizing a arbitrary cross section of the volume. Please refer to Appendix A for more details.

1.1.1 Input and Ideal Result

Our 3D CT data is provided by Roberts Research¹. It is a volume of the mouse's heart. The actual physical size of the data is very tiny, while the resolution of the data is very high ($585 \times 525 \times 892$).

Figure 1.1 and Figure 1.2 show the original data. The bright parts correspond to vessels and the dark parts to heart muscles and other injected material. Figure 1.1 shows four different slices of the original data. The bright white balls are corresponding to the cross sections of arteries and the small white balls are cross sections for vessels. Figure 1.2 show the whole data using *maximum intensity projection*.

Some of the features of this data are:

1. The size of the vessels varies from tiny capillaries to arteries
2. Small vessels have lower intensity while thicker ones have stronger intensity
3. The partial volume problem exists at smaller vessels as well as on the boundaries of bigger vessels
4. There are a number of artifacts of rings due to the reconstruction of the CT images
5. Random noise exists everywhere in the image

Figure 1.3b shows an example of the ideal result of this vessels.

¹<http://www.robarts.ca/>

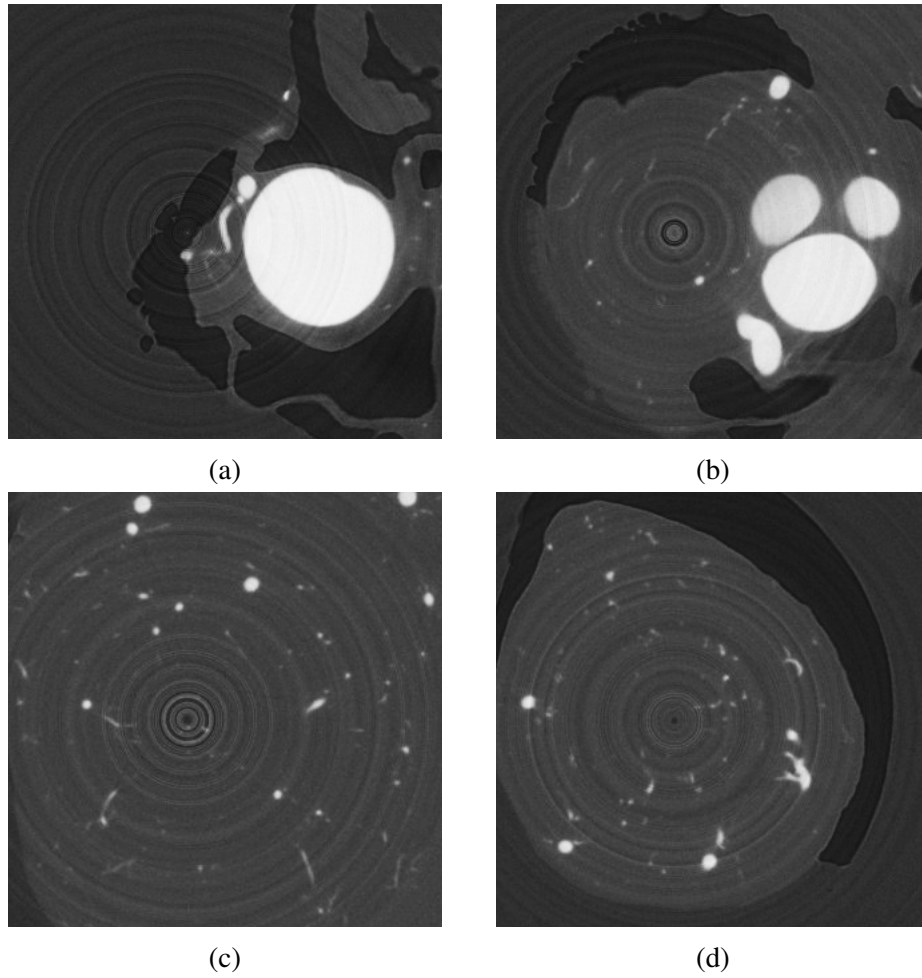
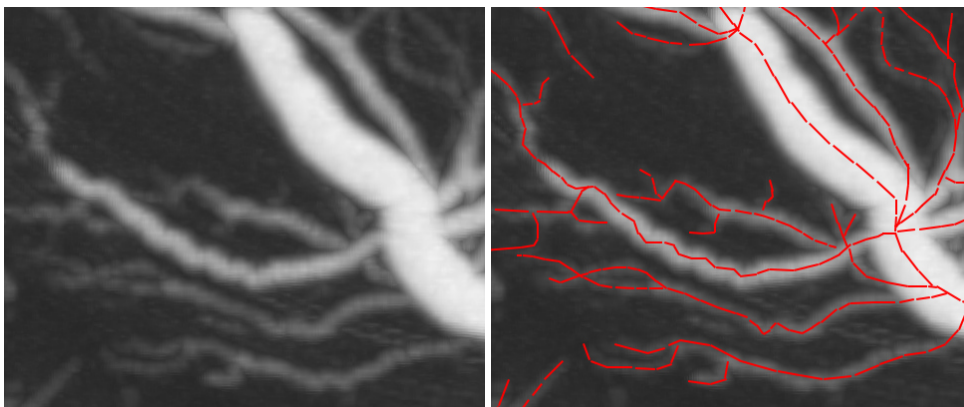


Figure 1.1: Four Slices of the Original Data



Figure 1.2: Original Data With Maximum Intensity Projection



(a) Original Data

(b) Ideal Result

Figure 1.3: Original Data and Ideal Result

1.1.2 Technical Problems

We are confronted with the three technical problems in order to get the result like those in Figure 1.3b: (A) *filtering (preprocessing)*, (B) *structure extraction*, (C) *topological constraints*.

Filtering (preprocessing) reduces different kinds of noise in the image. Structures can be either the segmentation or the centreline of the object. In this thesis, we are focused on extracting the centreline of the vessels. We conjecture that it is straight forward to get the segmentation given a correct centreline. Finally, topological constraints are enforced on the centrelines using *minimum spanning tree* algorithm.

(A) Filtering (Preprocessing)

There are two kinds of noise in our data — rings and random noise. Rings illustrated in Figure 1.4 are very common in CT images. They are concentric rings superimposed on the image while it is being scanned [24]. Rings are a structure noise in the following sense. For all points that are with the same distance to the centre of rings, the variation of intensity is similar. Figure 1.4 shows a dark ring and a bright ring in the image. Random noise is variation of intensity caused by the limitation of the digital sensors. Both of these noises are problematic; therefore image filters are required in order to remove the artifacts.

Ring Filter

The state-of-the-art ring filter was proposed by [40] using mean and median filtering. Rings artifacts are reduced to a great extent with this filter. Data before and after rings reduction are shown in Figure 1.5. More details about this filter is presented in Appendix B.

Vesselness Filter

Standard filters such as *Gaussian filter*, *mean filter*, or *median filter* are commonly used for non-structured white noise. They do not work well for our data because they smooth out the small vessels. We apply the vesselness filter [29] to our data. This filter can remove background noise while preserving structure details for small vessels. Figure 1.6 shows an image before and after vesselness filter. Section 2 is focused on vesselness filter. We sometime refer to it as “vesselness measure” and we use both of these two expressions in this thesis.

(B) Structure Extraction

There are two categories of methods to extract the structure of the object. Methods such as *graph cuts* segment the object of interest by labelling all image pixels into two subsets: object or background [6, 43, 25]. Some skeleton-based methods extract the centreline of the object [11, 14, 15, 28, 14]. Both extracting segmentation and extracting centreline are ill-posed problems since we don’t have a unique solution to either of the problems.

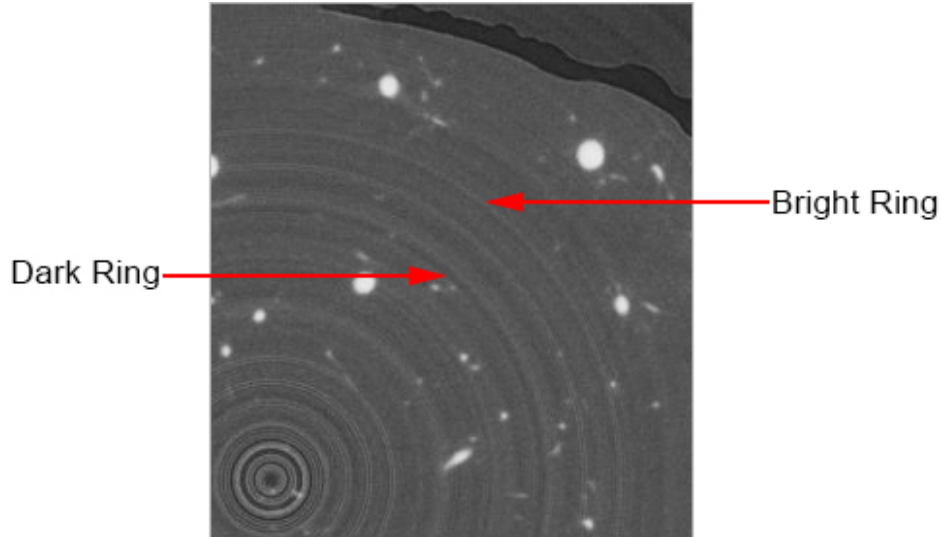


Figure 1.4: Rings Artifacts

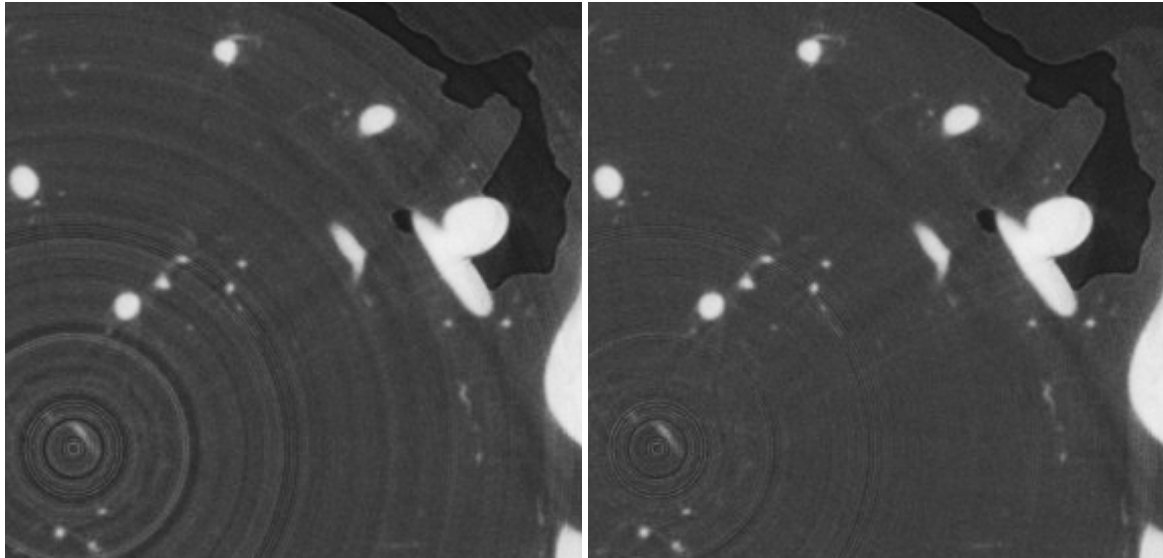
Segmentation

Image segmentation is the process of assigning different labels to image pixels according to their image attributes such as intensity, colour and etc. Binary segmentation labels the image into two subsets — foreground or background. One simple methods for image segmentation is thresholding. It segments an image as follows: if the intensity of a point is above the threshold, it is assigned to one label; otherwise it is assigned to the other label. Figure 1.7 shows an example of binary segmentation using thresholding. Rings as well as other image noise are picked up with a low threshold. The result is cleaner with a high threshold, but some of the small structures are lost. That is why thresholding does not work for our data and we need to use more advanced and sophisticated methods.

Centreline

Another way to analyze our data is to extract the centreline of vessels. The concept of centreline was first introduced by Blum et al [4]. It was originally referred to as the *topological skeleton* in [4]. It is nowadays also known as *medial* or *symmetric axes* [44]. A centreline is a continuous imaginary line through the centre of an object. Every point on the centreline must have more than one closest point to the boundary of the object.

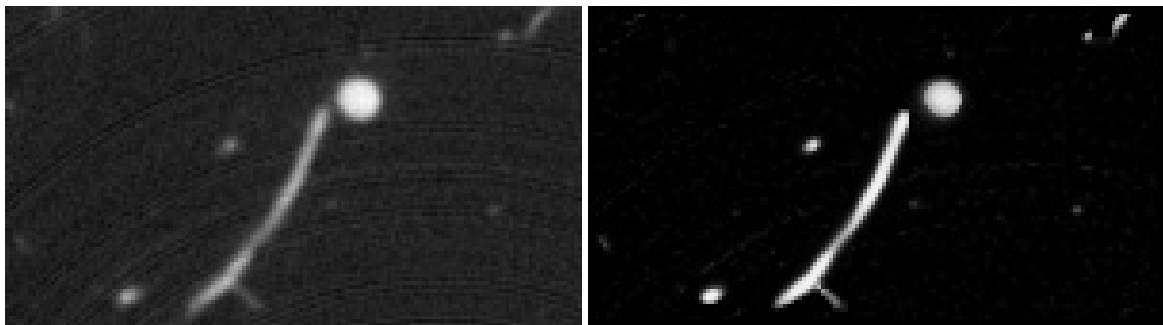
The actual representation of the centreline is sometime referred to as *discrete centreline* [44]. It can be represented in different ways. For example, it can be described as a set of independent points [44, 37, 28] (see Figure 1.8a). Centreline can also be represented as a set of line intervals (see Figure 1.8b). We define the *discrete centreline* as follows: a connected graph with certain properties for nodes (as being either pixels or line intervals) that are equidistant from multiple points on the object boundary. Typically, the graph that we are looking for is a



(a) Before Rings Reduction

(b) After Rings Reduction

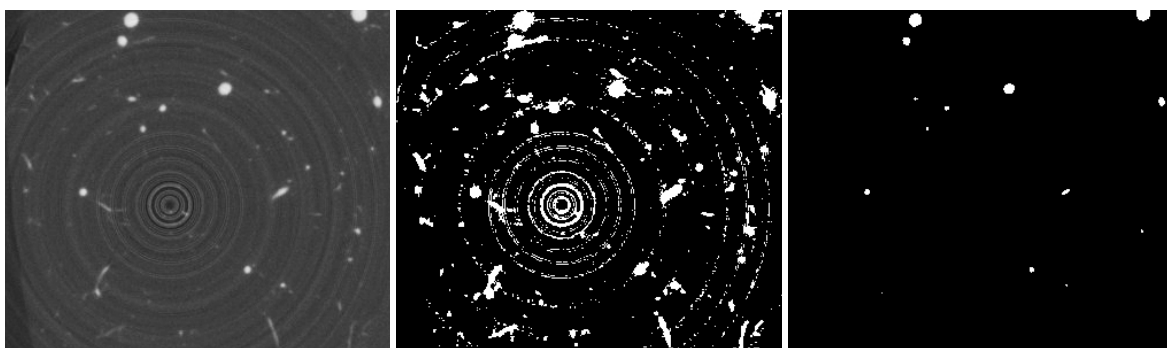
Figure 1.5: Rings Reduction



(a) Before Vesselness Filter

(b) After Vesselness Filter

Figure 1.6: Vesselness Filter



(a) Original Data

(b) Low Threshold

(c) High Threshold

Figure 1.7: Image Segmentation with Thresholding

tree (see more in Section (C) below). Figure 1.10b and Figure 1.11b shows some examples of discrete centrelines on real data.

For simplicity, we refer both *continuous centreline* and *discrete centreline* as *centreline* in the rest of this thesis.

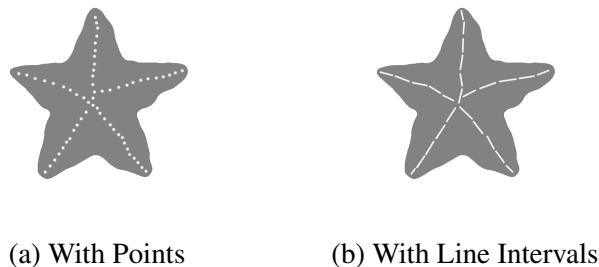


Figure 1.8: Discrete Centrelines

(C) Topological Constraints

To disambiguate our ill-posed structure extraction problems discussed in (B), different topological constraints can be enforced on the extracted structures. Two of the most important constraints are *connectivity constraint* and *tree-connectivity constraint*.

Connectivity Constraint

The connectivity constraint for segmentation ensure the following — there exists a path between any two points labelled as the same segment. Figure 1.9 shows an example of a segmentation without connectivity constraint. Notice that the fins of the birds are separated from the body.

A graph that satisfy a connectivity constraint can have loop. Figure 1.10c shows an example of enforcing connectivity constraint on the data points. The data points are pixels that are correlated to the centreline of the vessel. The green lines in Figure 1.10 are the connection between data points. Notice that we may have loops in the result with only connectivity constraint.

Tree-connectivity Constraint

A tree-connectivity constraint requires that the connective graph cannot have loops. Figure 1.10d show an example of enforcing the tree-connectivity constraint on the data points. Figure 1.11c shows an example of enforcing connectivity constraint on line intervals (red). The green lines indicate the connectivities.

Tree-connectivity constraint can be enforced using *minimum spanning tree* algorithm. See Chapter 4 for more details.



Figure 1.9: Segmentation Without Connectivity [43]

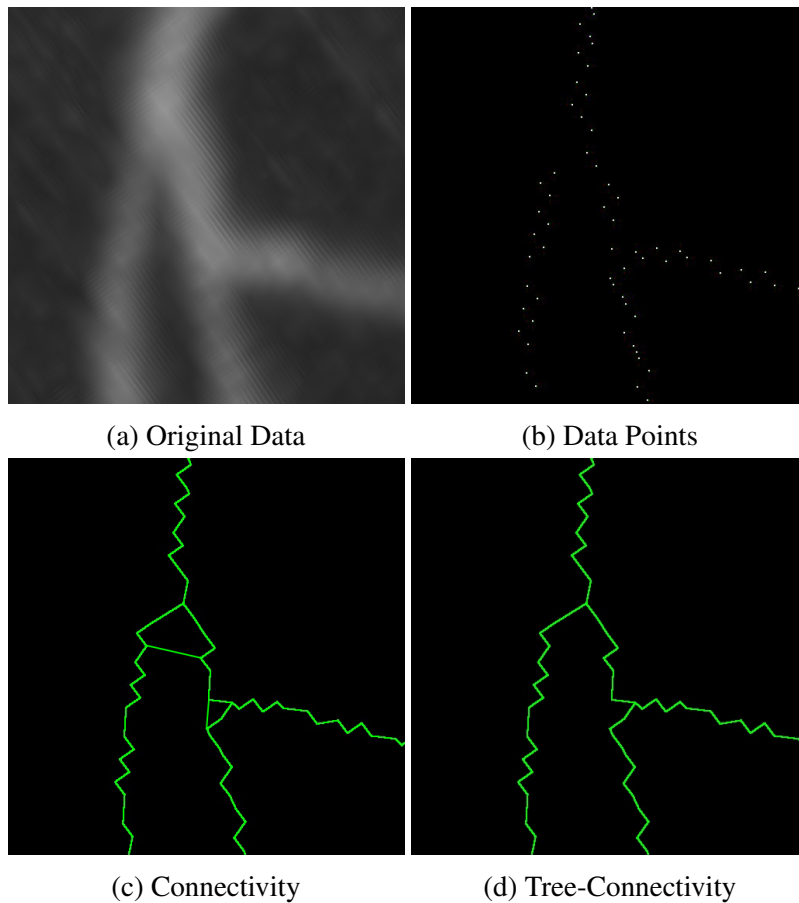


Figure 1.10: Centre Line With Data Points

1.1.3 Pipeline of The Algorithms

Figure 1.12 shows the pipeline of the algorithms. Our final goal is to extract the tree structures of the cardiac image. The tree constraint is enforced with a minimum spanning tree algorithm.

Before building a tree, we have to construct a connected graph. Vessel thinning and model fitting are two different methods of extracting the elements (or nodes) for the graph. Vessel thinning extract the voxels that are correlated with the centrelines of the vessels. Model fitting

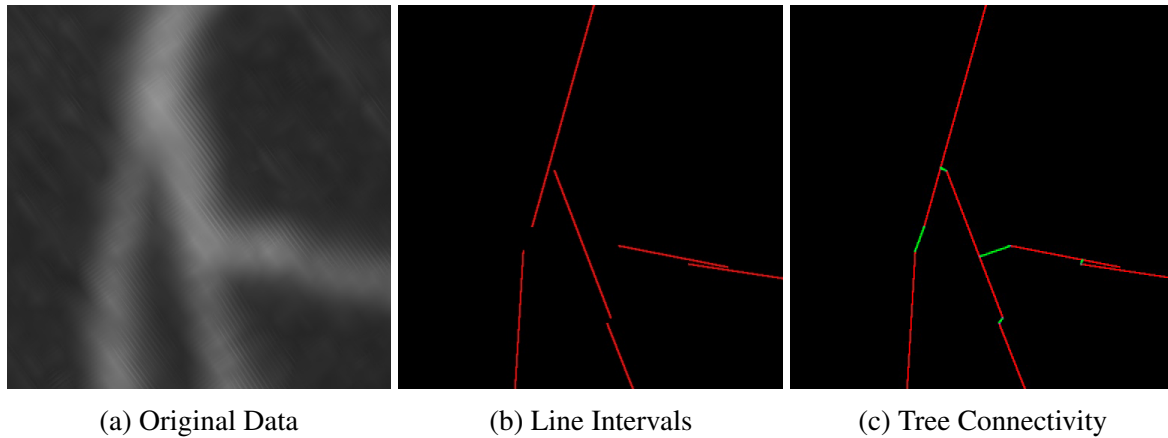


Figure 1.11: Centre Line With Line Intervals

fits line intervals to the vessels.

The first block of the pipeline is noise reduction. Two different filters are applied to the original data in order to remove moth rings artifacts and random noise.

1.2 Related Work

Thin structures are very common in medical image processing, and a lot of research has been done during the past decades. The research deals with at least one of the technical problems that we discussed in Section 1.1.2. The organization of this section is as the following.

Section 1.2.1 and Section 1.2.2 summarize related works about two different filters used in this thesis: *rings filter* and *vesselness filter*, which are related to Technical Problem (A).

Section 1.2.3 introduces *graph cuts* [6] which is focused on the segmentation of the object. Section 1.2.4 introduces some other methods, which are used to extract the centreline of the object. Topological constraints are enforced on both of these two types of methods. Section 1.2.3 and Section 1.2.4 are related to Technical Problems (B) and (C).

1.2.1 Rings Filtering

Rings artifacts are a number of concentric rings superimposed on the image while it is being scanned [24]. The presents of rings causes problem for post processing, such as noise reduction or image segmentation. Removing or reducing such artifacts is necessary and a lot of research has been done on that over the past decade.

Rings reduction can be done while the CT image is being scanned. They are referred to as pre-processing algorithms for rings reduction. Algorithms such as [1, 45, 32] are all pre-processing algorithms. Some other algorithms operate directly on the reconstructed images

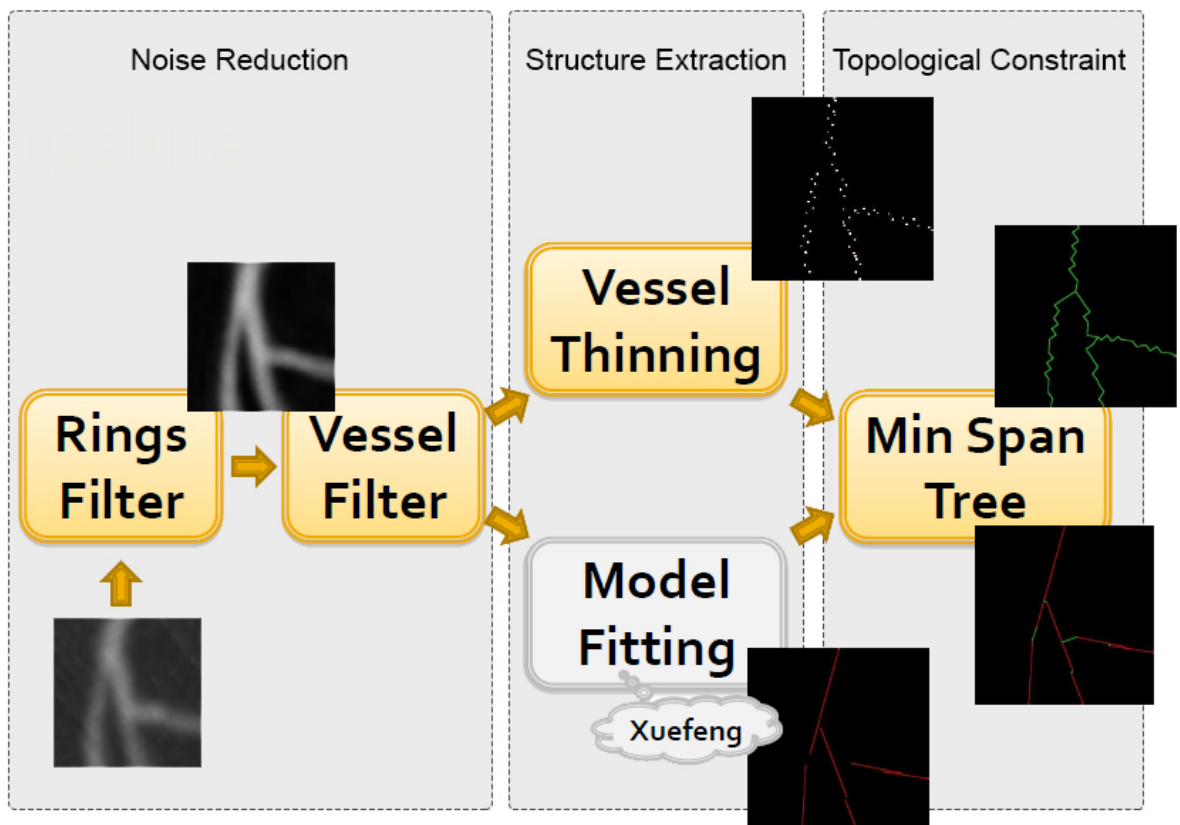


Figure 1.12: Pipeline of The Algorithms

[40, 2, 35, 27]. They are usually referred to as post-processing algorithms. In this thesis, we are only focused on the post-processing algorithms.

The state-of-the-art rings post-processing algorithm for rings reduction was initially proposed by Sijbers and Postnov [40]. The algorithm transforms the image from Cartesian coordinate to polar coordinate. Figure 1.13a shows an image with rings and Figure 1.13b shows its corresponding polar coordinate image. The problem of the ring artifacts in the original image becomes a problem of line artifacts in the polar coordinate image. And then a mean filter is applied to the image in polar coordinate. A artifacts template is generated by comparing the image before and after mean filter. The rings are corrected based on the artifacts template. Figure 1.13c shows the result after reducing the line artifacts in polar coordinate. Finally the image is transformed back into Cartesian coordinates.

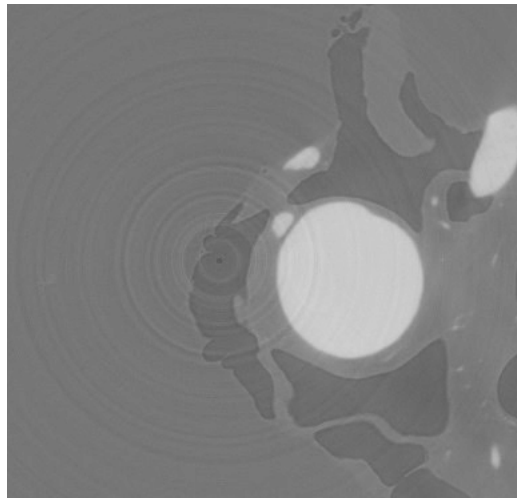
Many algorithms are based on the method described above such as Axelsson et al. [2]. However, the filtering does not necessary have to be done under polar coordinates. A similar algorithm in Cartesian coordinate is introduced by Prell et al. [35]. Some comparison of the ring filter under Cartesian coordinate and polar coordinates can be found in Prell et al. [35] and Kyriakou et al. [27].

1.2.2 Vesselness Filtering

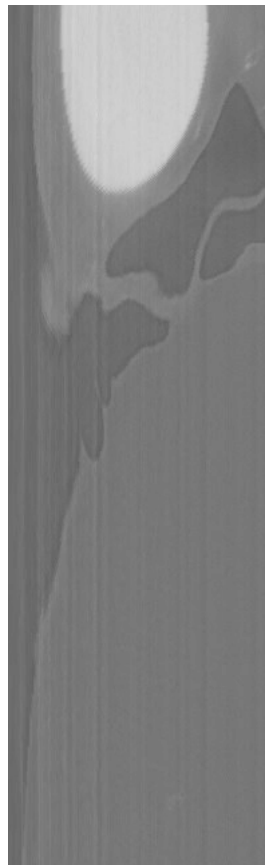
The vesselness filter was introduced by Frangi et al. [19]. It was initially called *vessel enhancement filter* in [19] because of the fact that this filter can reduce unexpected white noise in the image while preserving vessel structures. It is later referred to as *vesselness measure* [9, 10, 17] or *vesselness filter* [36, 18, 41]. In this thesis, we use both of the terminologies interchangeably.

The vesselness measure proposed by [19] calculate the measure indicating how likely a point belongs to a vessel. A critical steps is computing the Hessian matrix from the image data. This filter can also detect the major orientation of the vessels by computing the eigenvalue decomposition of the Hessian matrix. Figure 1.14 shows an example of vesselness measure computed from [19]. More details on this method are introduced in Chapter 2.

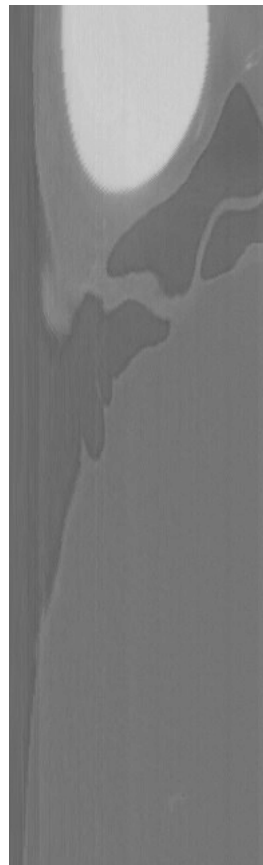
The vesselness filter is later used in many other applications. For example, it is used for detecting *space-time shapes* [3], for vessel segmentation [16, 7, 38], for detecting vascular connectivity [22] and etc.



(a) Cartesian Coordinate



(b) Polar Coordinates



(c) Polar Coordinate (Result)

Figure 1.13: Rings Reduction Method

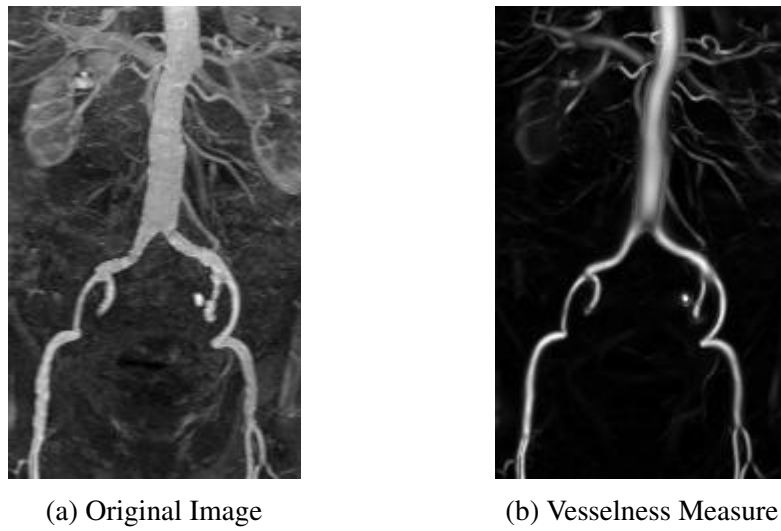


Figure 1.14: Vesselness Measure [19]

1.2.3 Graph Cut Segmentation

Graph cut has been widely use because of its capability in dealing with graph-based energy [6, 5]. It formulates the graph energy into the following:

$$E(f) = \sum_{p,q \in \mathcal{N}} V_{p,q}(f_p, f_q) + \sum_{p \in \mathcal{P}} D_p(f_p), \quad (1.1)$$

where $V_{p,q}(f_p, f_q)$ is the smooth cost for any neighbouring pixels p and q under a neighbourhood system \mathcal{N} ; and $D_p(f_p)$ is data cost for any pixel in the set of image pixels \mathcal{P} .

This smooth cost in graph cuts is for handling image noise. However, it has the an over-smoothing problem for thin structures. We refer to this as the over-smoothing problem. As is illustrated in Figure 1.15, the feet and the tentacles of the bee are lost with graph cuts segmentation method.

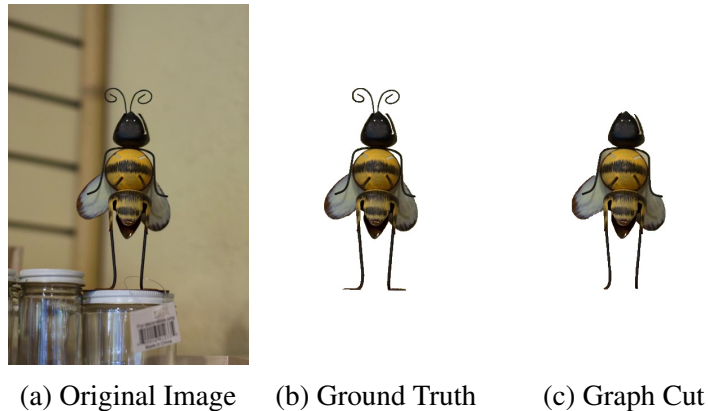


Figure 1.15: Over-smoothing of Thin Structure With Graph Cuts [25]

Multiple attempts have been made by previous researchers in order to address the over-smoothing problem. An attempt is through coupling edges in graph cuts [23, 26]. They achieve this by categorizing the image edges into groups and applying some discount function on graph cuts if some edges in the same group are cut. Take the image of the bee (Figure 1.15) as an example. The tentacles of the bee are thin structures. Therefore, the smooth cost is very high in order to segment the tentacles. But the boundary edges of the tentacles have similar appearance — dark on one side and bright on the other side. The edges along the boundary of the tentacles can be categorized as the same group. The smooth cost for these edges of the same group are reduced in order to segment thin structures such as the tentacles. The problem of this approach is that the categorization of the edges is not reliable. Therefore, some other image noise or artifacts are introduced to the final segmentation.

Some research has also been carried out in order to ensure connectivity constraint in graph

cuts. A interactive method is proposed by Vicente et al. [43]. This method first get an initial segmentation using graph cuts (Figure 1.16a). Then user can add additional input points and these points will be connected to the initial segmentation using *DijkstraGC algorithm*. Please refer to Vicente et al. [43] for more details about the *DijkstraGC algorithm*.

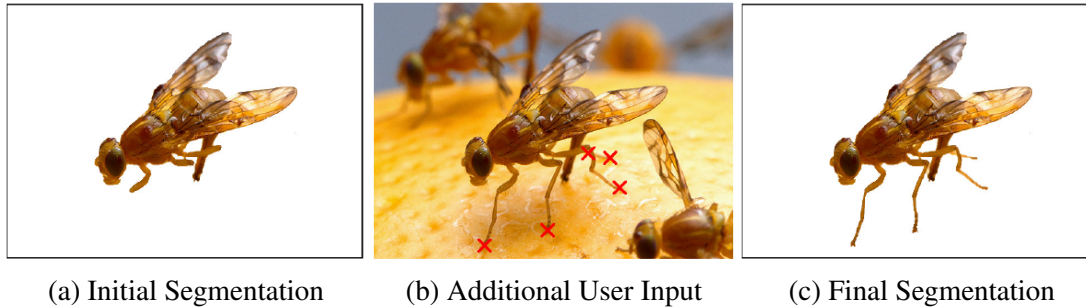


Figure 1.16: Connectivity Constraint in Graph Cut [43]

1.2.4 Centreline-Based Methods

The centreline of tubular structures are extracted by computing the minimal path between two user-input points [11, 14, 15]. It can detect global minimum of an active contour model's energy between two endpoints [11].

The benefits of the minimal path approach include global minimizers, fast computation and incorporation of user input. A drawback of this approach is that it represents the vessel with a curve which runs through the interior of the vessels instead of a full tubular surface. In order to overcome this, a fourth dimension of the vessel is introduced by [28], which is the radius of the vessel. Each point on the 4-D curve consists of 3 dimensional spatial coordinates plus a fourth dimension which describes the radius of the vessel at that corresponding 3-D point in space [28]. Thus, each 4-D point represents a sphere in 3-D space, and the vessel is obtained by taking the envelope of these spheres as we move along the 4-D curve. This approach takes into consideration both the mean and variances for sphere $sp = (p, r)$ in an image where p is 3D points and r is radius. Finally they compute the minimum path between two user input spheres. Please refer to Li and Yezzi [28] for more details.

Some of the results of the 4D path approach are shown in Figure 1.17. Notice that the connectivity constraint is automatically enforced when computing the minimum path between two user input points.

These centreline-based methods are normally applied for extracting centrelines of colons because a colon has only two endpoints. It is not applicable to vessels because the number of endpoints is enormous in the data and the endpoints of the vessels are mostly capillaries which

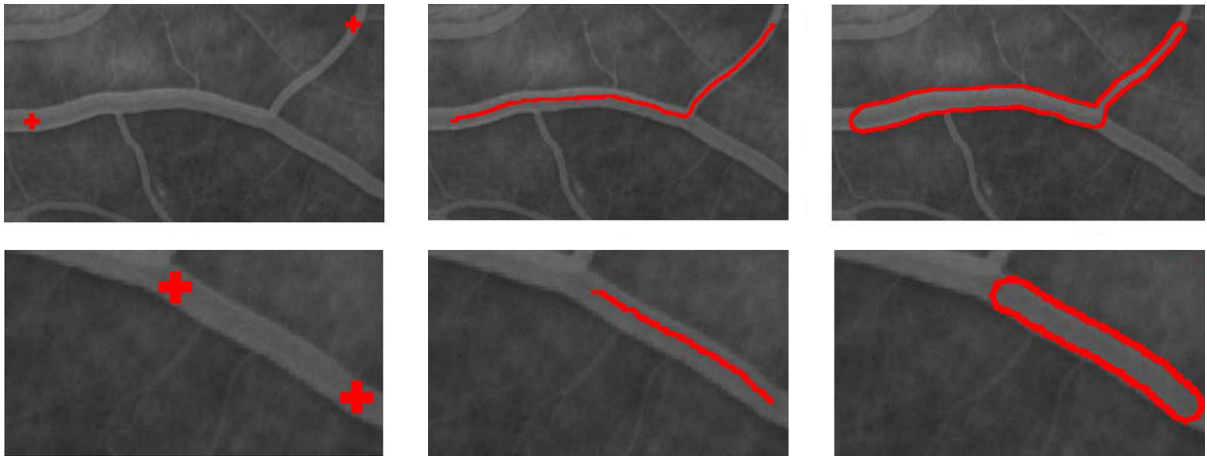


Figure 1.17: 4D Path Result [28]

are not easy to keep track of with human eyes.

1.3 Contributions

We gave a more intuitive explanation of the vesselness filter in Chapter 2. We explained in detail why this measure works for 3D tabular structures. And also we discussed about different ways of adjusting this filter so that it can be used for detecting other image structures (such as balls on 2D images).

We implemented and compared two different methods for extracting the centrelines of the vessels in Chapter 3. We developed a *vessel thinning* method inspired by *non-maximum suppression*. Our colleague Xuefeng Chang implemented the recent optimization-based model fitting algorithm. We shows an simpler model fitting problem in Section 3.3.2.

In Chapter 4, we implemented a *minimum spanning tree* algorithm and used it to enforced the tree-connectivity constraint on both of the two types of centrelines.

Finally, we implemented visualization tools in order to better analyze our data. We are able to visualize the *maximum intensity projection* and an arbitrary cross section of the 3D data.

1.4 Outline of the Thesis

Vesselness filter (or vesselness measure) is explained in detail in Chapter 2. The use of the *Harris conner detector* is very similar to the use of vesselness filter and it is presented in Section 2.2. The *Hessian matrix* is used for the vesselness filter and it is explained in Section 2.3 and Section 2.4. In Section 2.5 and Section 2.6, the *eigenvalues* of *Hessian matrix* are described.

Two different methods of extracting centreline of the vessel are explored in Chapter 3. This thesis is focused on the first method — *data thinning* in Section 3.2. A second method is mostly carried out by a colleague, Xuefeng Chang. Some brief introduction of model fitting is presented in Section 3.3.

The use of *minimum spanning tree* is explained in Chapter 4. Two commonly used algorithms for computing *minimum spanning tree* is presented in Section 4.1. The use of *minimum spanning tree* algorithm on our data is discussed in Section 4.2 and Section 4.3.

Notice that visualization is also a very important part of the project. Some of the implementations of visualization are explained in Appendix A. A rings filter is implemented according to [40] and it is briefly introduced in Appendix B. Appendix C introduces some well-known properties for eigenvelues and eigenvector that are used in Chapter 2. Appendix D introduce some basic knowledge about 3D geometry which is used in Chapter 4.

Chapter 2

Vesselness Measure

2.1 Overview

The vesselness measure intuitively describe the likelihood of a point being part of a vessel. The higher the value of the vesselness of a given point, the more likely it is vessel. The vesselness measure uses a combination of Gaussian filter and *Hessian matrix*, which was proposed in Frangi et al. [19]. This method first was introduced in 1998 and became a gold standard for vesselness measure ever since then. Some related work has been introduced in Section 1.2.2. The terminologies *vesselness measure* and *vesselness filter* are equivalent and we use them iteratively in this chapter.

It is not reliable to judge weather a point belongs to a vessel or not based the intensity of that point. The vesselness measure takes advantage of the following two properties for a vessel: (a) the intensity stays unchanged along the direction of a vessel; (b) the intensity varies a lot in the normal direction of a vessel. Vesselness measure is capable of aggregating the intensity information of neighbouring points using Gaussian filter. It can also be derived for the major orientation of the vessel by computing the eigenvalue decomposition of the Hessian matrix. As a result, it is very powerful in suppressing background noise.

We start this chapter with a *Harris corner detector* in Section 2.2. The Harris corner detector is a well-known image corner detector in Computer Vision. It is similar to vesselness filter and it can be easily derived geometrically. Explaining the Harris corner detector will help with the describing of vesselness filter. Harris corner detector and vesselness filter have the following similarities:

1. Both of them extract the major orientation of local image structures based on eigenvalue decomposition;
2. Both of them are using 3×3 matrices for 3D images (or 2×2 matrices for 2D images);

3. Both of them aggregate intensity information of neighbouring points.

We explain the vesselness filter in Section 2.3. Section 2.3 explains the vessels in 1D and 2D with respect to vessels in 3D. Hessian matrix is discussed in Section 2.4. We explain why we have to combine the Gaussian filter with Hessian matrix. Section 2.5 describe different ways of combining of the eigenvalues and explain why the current vesselness detector is being used. Section 2.6 explains how to decide the correct scale of the vessel.

2.2 Harris Corner Detector

The original idea of Harris corner detector was proposed by Harris and Stephens [20]. A very good derivation is available in Derpanis [13].

2.2.1 Derivation of The Matrix for Harris Detector

The main goal of the Harris corner detector is detecting the shape corners in an image. It is impossible to tell whether this point belongs to a corner or not based on the intensity of one point. That's why Harris corner detector makes judgement based on a set of points within a certain windows W . The sum (or weighted sum) of the intensities within a certain window is computer. If the shifting of a window W in any direction would give a large change in intensity, then a corner exists at that position. The change of intensity for the shift $s = [\Delta x \ \Delta y]^T$ is:

$$d(\Delta x, \Delta y) = \sum_W w(x, y) [I(x + \Delta x, y + \Delta y) - I(x, y)]^2,$$

where $w(x, y)$ is a weight function. That is either a rectangular function $\omega = 1$ or a Gaussian weighting function $\omega = e^{-(x^2+y^2)/(2\sigma^2)}$.

The shift image intensity is approximated by a Taylor expansion truncated to the first order terms,

$$\begin{aligned} I(x + \Delta x, y + \Delta y) &\approx I(x, y) + I_x \Delta x + I_y \Delta y \\ &= I(x, y) + s^T \cdot \nabla I, \end{aligned}$$

where $\nabla I = [I_x \ I_y]^T$ is the gradient of the image intensity and $s = [\Delta x \ \Delta y]^T$.

Therefore,

$$\begin{aligned}
d(\Delta x, \Delta y) &= \sum_w w(x, y) [I(x_i + \Delta x, y_i + \Delta y) - I(x_i, y_i)]^2 \\
&\approx \sum_w w(x, y) [s^T \nabla I]^2 \\
&= \sum_w w(x, y) [s^T \nabla I \nabla I^T s] \\
&= s^T \left(\sum_w w(x, y) \nabla I \nabla I^T \right) s
\end{aligned} \tag{2.1}$$

Let $M(x, y)$ be a 2×2 matrix computed from image derivatives,

$$M(x, y) = \sum_w w(x, y) \cdot \nabla I \cdot \nabla I^T = \sum_w w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \tag{2.2}$$

Then Equation (2.1) can be further written as,

$$d(\Delta x, \Delta y) = s^T M(x, y) s. \tag{2.3}$$

2.2.2 Eigenvalues and Eigenvectors of Harris Detector

Let λ_i the i^{th} eigenvalue of matrix $M(x, y)$ and v_i be the corresponding eigenvectors. Based on the definition of eigenvalues and eigenvectors, we have,

$$M(x, y)v_i = \lambda_i v_i.$$

We can left multiply v_i^T on both sides giving:

$$v_i^T M(x, y)v_i = \lambda_i v_i^T v_i, \tag{2.4}$$

If v_i is a unit vector ($v_i^T v_i = 1$), then Equation (2.4) can be written as,

$$v_i^T M(x, y)v_i = \lambda_i. \tag{2.5}$$

Comparing Equation (2.5) and Equation (2.3), it is not hard to see the geometric meaning of eigenvalues — the eigenvalue λ_i describe the variation of image intensity along direction v_i .

It can be proofed that λ_i is real number rather than complex (Theorem C.0.1) and the eigenvectors are always perpendicular to each other (Appendix Theorem C.0.2).

Because of the fact that the eigenvalues of $M(x, y)$ are independent to the choice of $s = [\Delta x \ \Delta y]^T$, therefore, they form a rotationally invariant description of the image properties at

the current position (x, y) . And these properties are,

- $\lambda_1 \approx 0, \lambda_2 \approx 0$
If both λ_1, λ_2 are small, the windowed image region is of approximately constant intensity.
- $\lambda_1 \approx 0, \lambda_2 \gg 0$
If one eigenvalue is high and the other low, only local shifts in one direction (along the ridge) cause little change in $d(x, y)$ and significant change in the orthogonal direction. This indicates an edge.
- $\lambda_1 \gg 0, \lambda_2 \gg 0$
If both eigenvalues are high, then shift in any direction results in a significant increase. This indicates a corner.

2.2.3 Visualization of Eigenvalues and Eigenvectors

$M(x, y)$ can be represented via the following ellipsoid:

$$s^T M(x, y) s = 1. \quad (2.6)$$

Eigenvalue decomposition gives:

$$M(x, y) = U \Lambda U^T,$$

where Λ is a diagonal matrix and U is a rotation matrix:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \text{ and } U = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

Applying this to Equation (2.6) we have,

$$s^T U \Lambda U^T s = (U^T s)^T \Lambda (U^T s) = 1$$

By rotating the coordinates from s to s' through $s' = U^T s = [\Delta x', \Delta y']^T$, we get $s'^T \Lambda s' = 1$. That is,

$$\lambda_1 \Delta x'^2 + \lambda_2 \Delta y'^2 = \frac{\Delta x'^2}{\left(\frac{1}{\sqrt{\lambda_1}}\right)^2} + \frac{\Delta y'^2}{\left(\frac{1}{\sqrt{\lambda_2}}\right)^2} = 1 \quad (2.7)$$

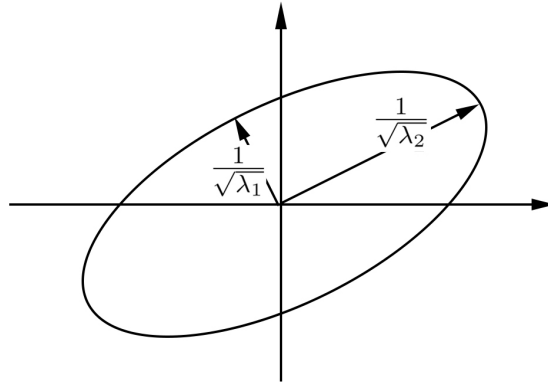


Figure 2.1: Visualized Eigenvalues with Ellipsoid

The semi-principal axes of the ellipsoid are $\frac{1}{\sqrt{\lambda_1}}$, and $\frac{1}{\sqrt{\lambda_2}}$. It can be visualized as Figure 2.1.

2.2.4 Combination of Eigenvalues

A proper combination of the eigenvalues is needed in order to get a descriptive corner measure. The Harris corner detector looks for the feature with both $\lambda_1 \gg 0$ and $\lambda_2 \gg 0$. There are many options including but limited to the following.

1. In the original paper, Harris and Stephen [20] use the following measure (with the value of κ to be determined empirically),

$$\begin{aligned} \mathcal{R} &= \lambda_1 \lambda_2 - \kappa \cdot (\lambda_1 + \lambda_2)^2 \\ &= \det(M) - \kappa \cdot \text{trace}^2(M). \end{aligned}$$

2. Shi and Tomasi [39] compute the minimum of the eigenvalues as the corner feature response,

$$\mathcal{R} = \min(\lambda_1, \lambda_2).$$

3. Noble's [33] corner measure computes the harmonic mean of the eigenvalues

$$\mathcal{R} = 2 \frac{\det(M)}{\text{trace}(M) + \epsilon}.$$

2.3 Conceptual Vessels in Different Dimensions

It is easier to describe the vesselness filter in 1D and 2D and then upgrade it to a higher dimension. Therefore, we describe the corresponding shapes in 2D and 1D images for a 3D vessels. We will discuss why the downgrading is reasonable and what information is preserved or lost during the downgrading.

The degree of freedom of shapes in different dimensions are summarized in Table 2.1.

Dimension	Shape Representation	Degree of Freedom
3D	Tube	7
2D	Rectangle	5
2D	Ball	3
1D	Box Function	2

Table 2.1: Corresponding Shapes for Vessels in 3D, 2D and 1D

2.3.1 3D Vessels as Tubes

In 3D, vessel can be thought of as a set of 3D tubes. Each tube have two end points (6 degrees of freedom) and one radius (1 degree of freedom); therefore there are 7 degrees of freedom in total.

2.3.2 2D Vessels as Rectangles and Balls

There are two different ways to project a 3D vessel onto a 2D plane.

If the projection plane is parallel to the orientation of the vessel, the vessels are projected as rectangles (Figure 2.2a). There are 5 degrees of freedom for a rectangle — two endpoints (4 degrees of freedom) and a radius (1 degree of freedom).

If the projection plane is perpendicular to the orientation of the vessel, the projection of the vessels become balls (Figure 2.2b). There are only 3 degrees of freedom for a ball — two for position and one for radius.

Degrading the 3D tubes to 2D rectangles can preserve the orientation information of vessels. However, the orientation can be handled with Hessian matrix easily (as is shown later in Section 2.4). It is the distance to the centreline of the tubes that matters. Therefore, it is also reasonable to degrade the 3D tubes to 2D balls. We will describe Hessian matrix for both cases in Section 2.4.

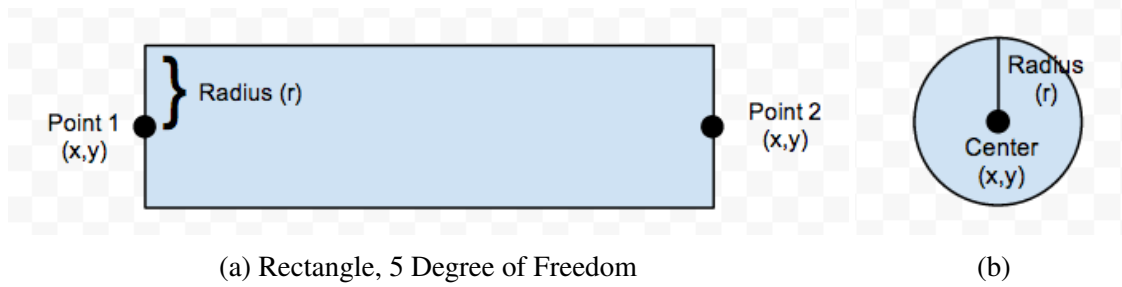


Figure 2.2: Conceptual Vessels in 2D

2.3.3 1D Vessels as Box Functions

The projection of 2D vessel (rectangle) along the orientation of the vessel is a 1D box function as the following,

$$f(x) = \begin{cases} C & \text{if } |x - \mu| < r \\ 0 & \text{otherwise} \end{cases} \quad (2.8)$$

where C is a constant, μ is the center of the box function and r is the size (or radius) of the box function. A 1D box function has 2 degrees of freedom.

2.4 Hessian Matrix and Gaussian Derivatives

Hessian matrix and 2^{nd} derivative of Gaussian filter are the two most important concepts used for vesselness filter. We first explain the use of the 2^{nd} derivative of Gaussian to detect the box functions in a 1D image (Section 2.4.1). And then the Hessian matrix is combined with Gaussian in order to detect 2D balls and rectangles in the images (Section 2.4.3 and Section 2.4.2). The detection of 3D tubes (or vessels in 3D) are introduced in Section 2.4.4.

2.4.1 Second Derivative of Gaussian in 1D

Assume that we have a 1D image, which can be described as a 1D discrete function. This 1D image contains some box functions with unknown centres and radii (Figure 2.3). The goal is to detect the centre of the box functions as well as their radii.

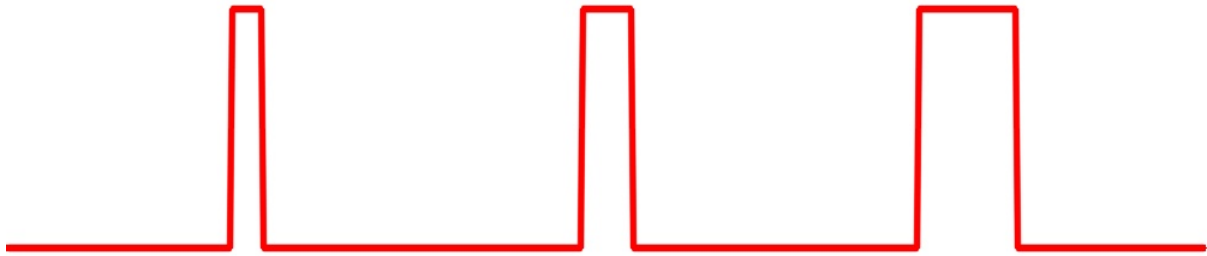


Figure 2.3: 1D Box Function

We use the second derivative of the Gaussian function to achieve this. The equations for Gaussian, first derivative of Gaussian, and second derivative of Gaussian are shown as follows.

- Gaussian

$$\mathcal{G}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Derivative of Gaussian

$$\frac{\partial \mathcal{G}(x)}{\partial x} = -\frac{x-\mu}{\sqrt{2\pi}\sigma^3} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Second derivative of Gaussian

$$\frac{\partial^2 \mathcal{G}(x)}{\partial x^2} = \frac{(x-\mu)^2 - \sigma^2}{\sqrt{2\pi}\sigma^5} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.9)$$

The equations are plotted in Figure 2.4.

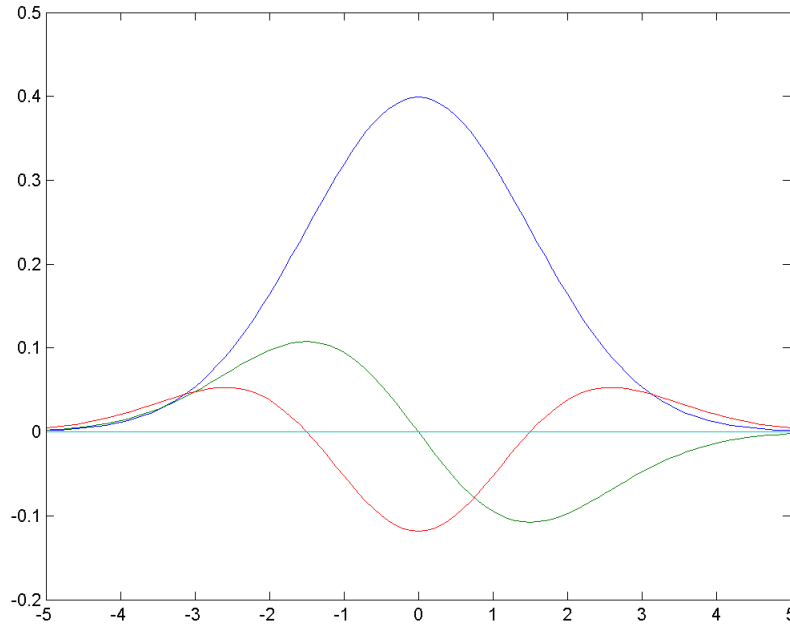


Figure 2.4: Gaussian (Blue), 1st and 2nd derivative of Gaussian (Green, Red)

For the second derivative of Gaussian Equation (2.9), notice that it is smaller than zero within $[-\sigma + \mu, \sigma + \mu]$, and greater than zero elsewhere.

Assume that μ and σ of the 2nd derivative of Gaussian matches the centre and radius of a box function. In another word, $f(x)$ is constant C within $[-\sigma + \mu, \sigma + \mu]$ and $f(x) = 0$ otherwise. The convolution of this box function with the second order derivative of Gaussian is,

$$\begin{aligned}
\int_{-\sigma+\mu}^{\sigma+\mu} \frac{\partial^2 \mathcal{G}(x)}{\partial x^2} f(x) dx &= C \cdot \left. \frac{\partial \mathcal{G}(x)}{\partial x} \right|_{-\sigma+\mu}^{\sigma+\mu} \\
&= -\frac{C(x-\mu)}{\sqrt{2\pi}\sigma^3} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \Big|_{-\sigma+\mu}^{\sigma+\mu} \\
&= -\frac{C(\sigma+\mu-\mu)}{\sqrt{2\pi}\sigma^3} e^{-\frac{(\sigma+\mu-\mu)^2}{2\sigma^2}} + \frac{C(-\sigma+\mu-\mu)}{\sqrt{2\pi}\sigma^3} e^{-\frac{(-\sigma+\mu-\mu)^2}{2\sigma^2}} \quad (2.10) \\
&= -\frac{2C\sigma}{\sqrt{2\pi}\sigma^3} e^{-\frac{1}{2}} \\
&= -\sqrt{\frac{2}{e\pi}} \frac{C}{\sigma^2}.
\end{aligned}$$

Notice that if we have a positive box function with $f(x) = C$ greater than 0 within the box, the result of the convolution is a negative value. We refer to the absolute value of the result of convolution as the response.

If and only if μ and σ of the 2nd derivative of Gaussian matches the centre and radius of a box function, the convolution generate a highest response. This argument is illustrated by Figure 2.5. In the Figure 2.5, the red lines represent the box function and curves represent a second order of derivative of Gaussian. We draw the negative of the second derivative of Gaussian for better visualization. In Figure 2.5a, all three second derivative of Gaussian have the same mean value μ . Sigma of the blue Gaussian is the same with the radius of the box function, while sigma of the purple and green one are smaller and bigger respectively. In Figure 2.5b, all three second derivative of Gaussian have the same variance σ . The blue one is consistent with the box function. The other two are sifted to the right and the left a little bit.

For example, the three boxes in Figure 2.3 are with the radii of 6 pixels, 10 pixels and 20 pixels respectively. We compute the convolution with the second derivative of Gaussian with the images for all image position μ and three different sigmas $\sigma = 6$, $\sigma = 10$ and $\sigma = 20$, which matches the radii of the box functions respectively. The responses are displayed in Figure 2.6a-c. Notice that whenever the sigma of 2nd derivate of Gaussian matches the radii of the box function, we have the best response. For example, with $\sigma = 6$ (Figure 2.6b), the best response is at the centre of the first box function.

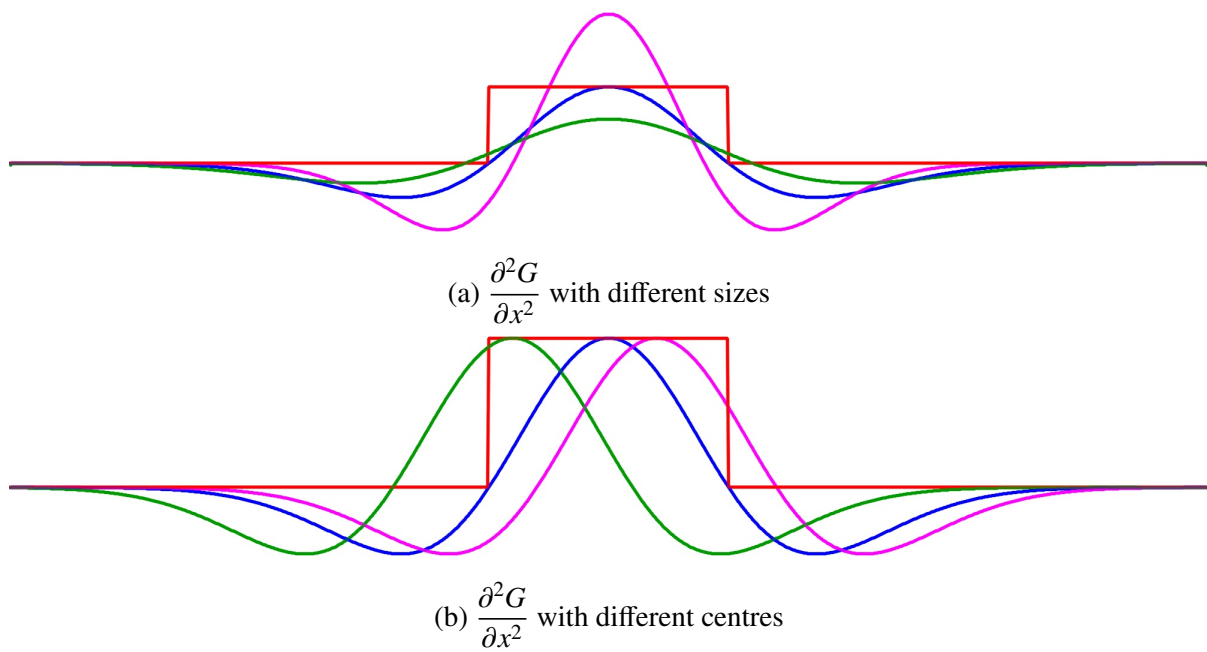


Figure 2.5: Find the best match of 2^{nd} derivative of gaussian $\frac{\partial^2 G}{\partial x^2}$ for box function

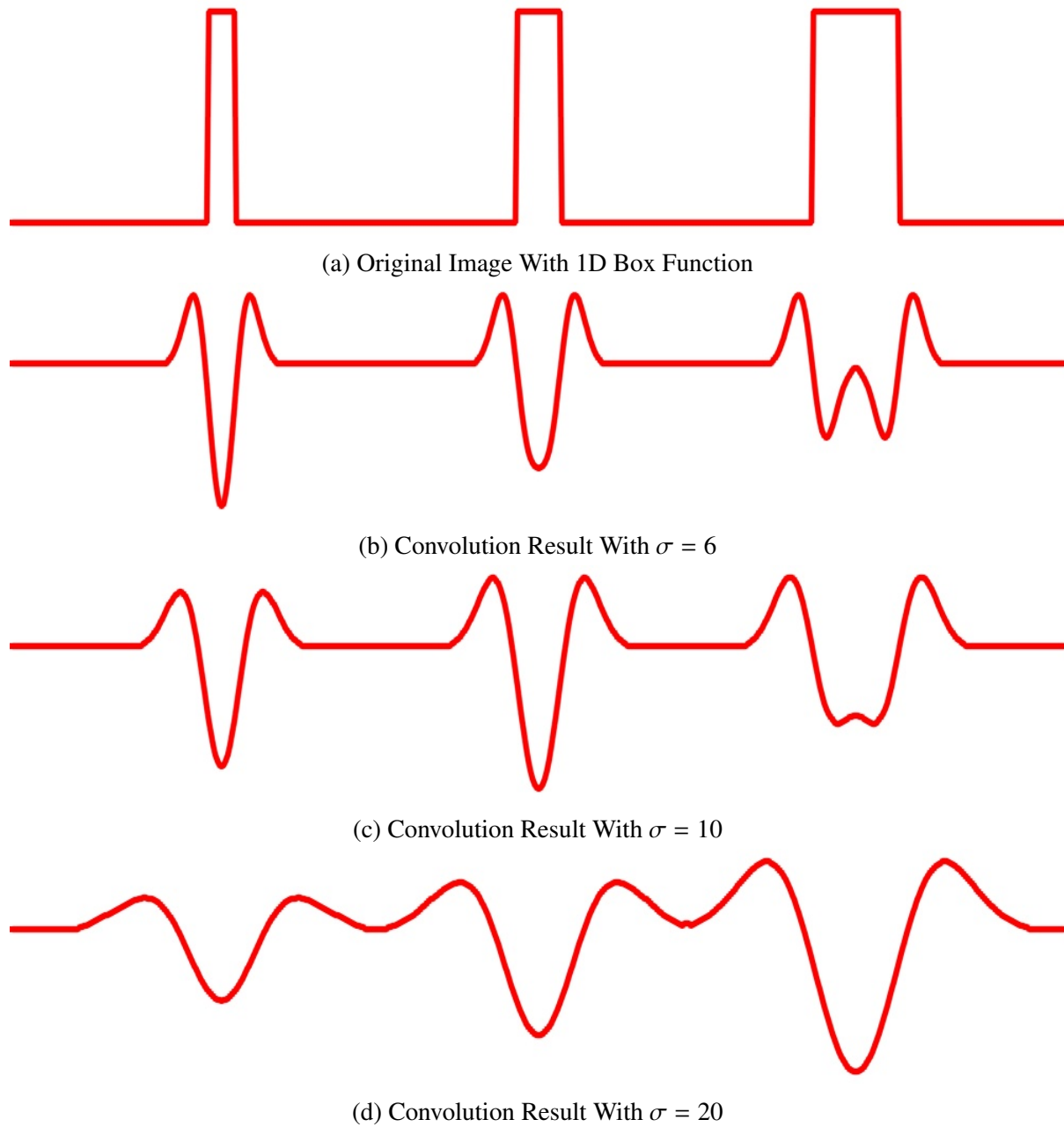


Figure 2.6: Convolution of 1D Box Function With 2^{nd} Derivative of Gaussian

2.4.2 Hessian for Rectangles in 2D

We need to consider the orientations of vessels in 2D images, which can also be considered as rectangles (see Section 2.3.2). One very intuitive way to deal with this problem is to use filters with different orientations. However, filters with multiple orientations are hard to design and the running speed can be slow.

In Section 2.2, the use of Harris corner detector is invariant to orientation. Is there a filter that can measure the vessel and is invariant to vessel orientation? We can achieve this with Hessian matrix.

The Hessian matrix in 2D is given by,

$$\mathcal{H}(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix},$$

where f is 2D discrete function. Each entry of the Hessian matrix is a second derivative of function f .

In order to use the Hessian matrix for vessel detection, we need to use it along with a Gaussian filter \mathcal{G} . We first blur our image \mathcal{I} with the Gaussian filter and then compute the Hessian matrix on the blurred image $\mathcal{H}(\mathcal{G} * \mathcal{I})$. The benefit of doing this is illustrated in rest of this section.

Figure 2.7 shows an example of 2D vessel as 3 bright rectangles. All these three rectangles are aligned with the y axis, which indicates that the image does not have a gradient along the y direction; Therefore, the following entries in the Hessian matrix are zero

$$\mathcal{H}_{12}(f) = \frac{\partial^2 f}{\partial x \partial y} = 0$$

$$\mathcal{H}_{21}(f) = \frac{\partial^2 f}{\partial y \partial x} = 0$$

$$\mathcal{H}_{22}(f) = \frac{\partial^2 f}{\partial y^2} = 0$$



Figure 2.7: 2D Vessels

And the value of \mathcal{H}_{11} is

$$\begin{aligned}
 \mathcal{H}_{11}(f) &= \mathcal{H}_{11}(\mathcal{G} * \mathcal{I}) \\
 &= \frac{\partial}{\partial x} * \frac{\partial}{\partial x} * \mathcal{G} * \mathcal{I} = 0 \\
 &= \frac{\partial^2 \mathcal{G}}{\partial x^2} * \mathcal{I}
 \end{aligned}$$

\mathcal{H}_{11} is the convolution of the second derivative of Gaussian Equation (2.9) with our image, which is used for the 1D box function in Section 2.4.1. Based the previous discussion, if the sigma of the Gaussian matches the size of the vessel, the highest response is generated at the centre of the vessel.

The radii for the vessels in Figure 2.7 are 6 pixels, 10 pixels, and 20 pixels respectively. Figure 2.8a show the centre row that \mathcal{H}_{11} is computed with green dash lines. The sigmas of Gaussian filters that used for Figure 2.8b-d are 6 pixels, 10 pixels, and 20 pixels respectively. Notice that we always have the best response when the sigma of the Gaussian filter matches the vessel size.

If the orientation of the vessel is aligned with the x axis, the image gradient along the x axis is zero. The only non-zero entry in the Hessian matrix is \mathcal{H}_{22} . Similarly, the best response of the convolution is

$$\mathcal{H}_{22} = \frac{\partial^2 \mathcal{G}}{\partial y^2} * \mathcal{I}$$

How about a vessel with an arbitrary orientation? We can address this by computing the eigenvalues and eigenvectors for the Hessian matrix. Let λ_1 and λ_2 be the eigenvalues of the Hessian matrix \mathcal{H} , and the corresponding eigenvectors be v_1 and v_2 . The Hessian matrix can be decomposed into the following form using eigenvalue decomposition.

$$\mathcal{H} = U \Lambda U^T$$

where Λ is a diagonal matrix and U is a rotation matrix,

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \text{ and } U = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}.$$

This implies that we can rotate the image using a rotation matrix U so they align with the axis. The geometric meaning of the eigenvalue λ_i is the convolution of the image with a second order derivative of Gaussian on the direction of v_i .

Intuitively, if a pixel is close to the centreline of a vessel, it should satisfy the following two properties:

1. one of the eigenvalues λ_1 should be very close to zero;
2. the absolute value of the other eigenvalue should be a lot greater than zero $\lambda_2 \gg 0$.

Therefore, if we sort the eigenvalues based on their absolute values so that

$$|\lambda_1| < |\lambda_2|,$$

we can use the absolute value of second eigenvalue $|\lambda_2|$ as a intuitive measure about how close a pixel is to the centre of the vessel.

The first eigenvalue $|\lambda_1|$ should be close to zero. This implies that the convolution of a second derivative of Gaussian with the image along the direction of v_1 , we get zero. That simply means that the intensity of the image stay constant along the direction of v_1 . Therefore, the direction of v_1 indicates the major orientation of the vessel. Based on Appendix C.0.2, we know that v_2 is the normal of the vessel.

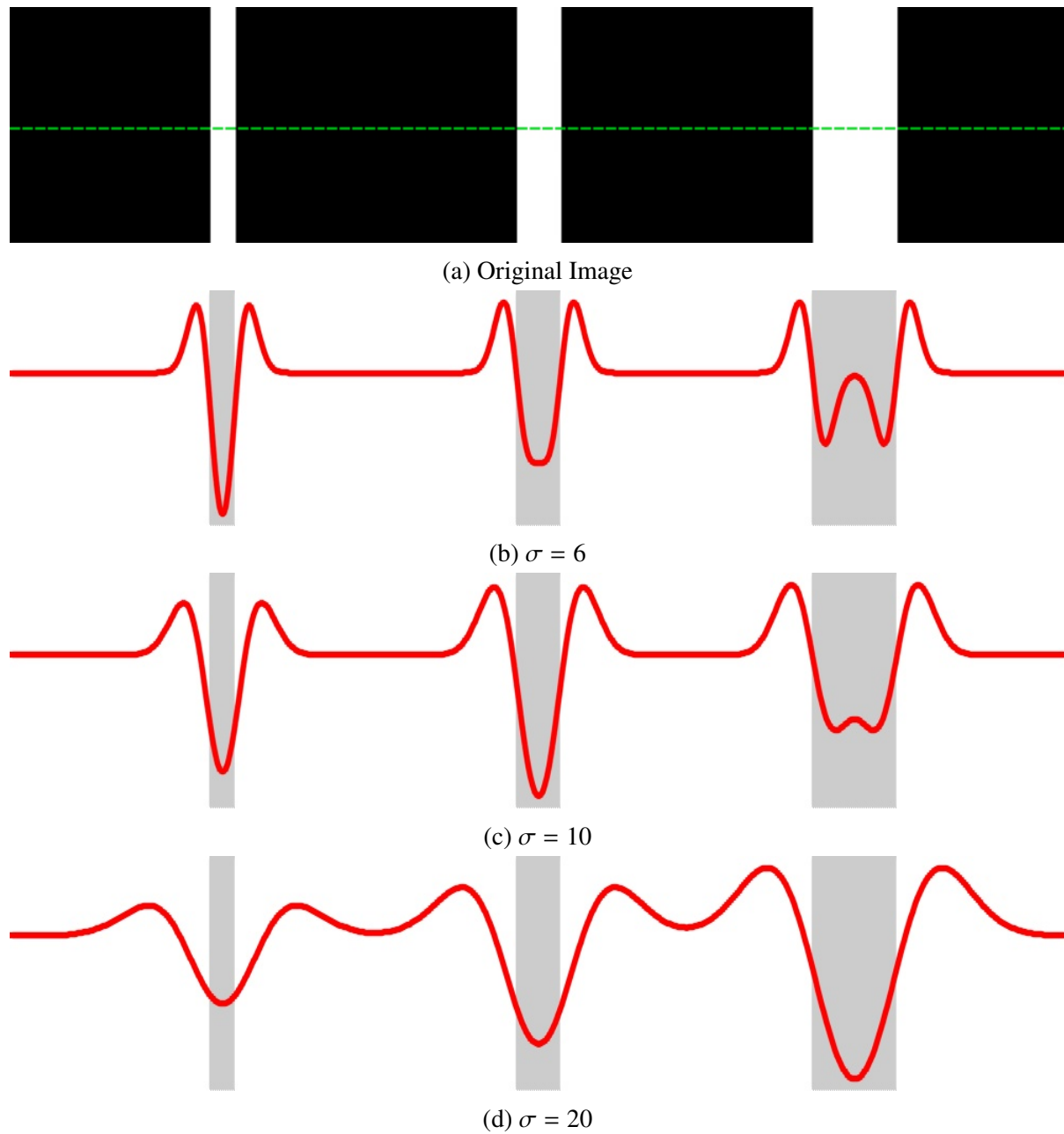


Figure 2.8: Eigenvalues of The Hessian Matrix for 2D Vessels

2.4.3 Hessian for Balls in 2D

The cross section of 3D vessels are 2D balls. Figure 2.9 gives us an example.

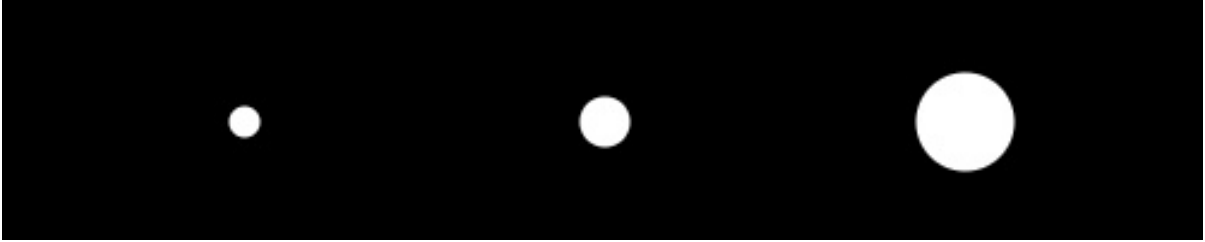


Figure 2.9: 2D Balls

In the Section 2.4.2, the intensity of the image stays constant along the orientation of the vessel. Blurring the image with Gaussian filter along the orientation of the vessel won't have any effects.

For balls, we need to look into the properties of the 2D Gaussian functions. The relationship between the Gaussian in 2D and 1D is shown in the following equations. Figure 2.10a shows an example of the first derivative of 2D Gaussian. Figure 2.10b shows an example of the second derivative of 2D Gaussian.

- Gaussian

$$\mathcal{G}(x, y) = \mathcal{G}(x) \cdot \mathcal{G}(y)$$

- Derivative of 2D Gaussian

$$\frac{\partial \mathcal{G}(x, y)}{\partial x} = \frac{\partial \mathcal{G}(x)}{\partial x} \cdot \mathcal{G}(y)$$

$$\frac{\partial \mathcal{G}(x, y)}{\partial y} = \frac{\partial \mathcal{G}(y)}{\partial y} \cdot \mathcal{G}(x)$$

- Second derivative of 2D Gaussian

$$\frac{\partial^2 \mathcal{G}(x, y)}{\partial x^2} = \frac{\partial^2 \mathcal{G}(x)}{\partial x^2} \cdot \mathcal{G}(y)$$

$$\frac{\partial^2 \mathcal{G}(x, y)}{\partial y^2} = \frac{\partial^2 \mathcal{G}(y)}{\partial y^2} \cdot \mathcal{G}(x)$$

$$\frac{\partial^2 \mathcal{G}(x, y)}{\partial x \partial y} = \frac{\partial \mathcal{G}(x)}{\partial x} \cdot \frac{\partial \mathcal{G}(y)}{\partial y}$$

To make things easy, we move both the centre of the Gaussian and the ball to the origin of

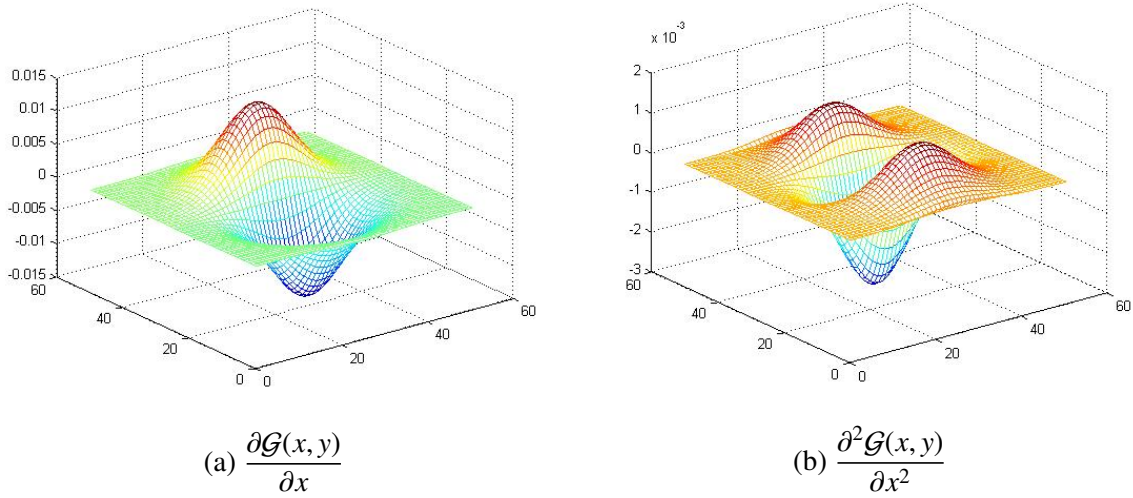


Figure 2.10: First and Second Derivative of Gaussian in 2D

our coordinate system. The equations for the Gaussian filter and the ball is show as:

$$\mathcal{G}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (2.11)$$

$$\mathcal{F}(x, y) = \begin{cases} C & \text{if } x^2 + y^2 < r^2 \\ 0 & \text{otherwise} \end{cases}. \quad (2.12)$$

Now the Gaussian only have one parameter — the variance σ . And Ball function also has only one parameter — the radius of the ball r . The convolution of the Equation (2.11) and Equation (2.12) gives,

$$\begin{aligned} \mathcal{R}(\sigma, r) &= \iint \frac{\partial^2 \mathcal{G}(x, y)}{\partial x^2} \cdot \mathcal{F}(x, y) dx dy \\ &= -C \cdot \frac{r^2}{2\sigma^4} e^{-\frac{r^2}{2\sigma^2}}. \end{aligned} \quad (2.13)$$

The result of the convolution is related to both the radius of the ball (r) and the variance of the Gaussian function (σ). We can get the extreme function value by taking the partial derivative of $\mathcal{R}(\sigma, r)$ over r :

$$\frac{\partial \mathcal{R}(\sigma, r)}{\partial r} = \left(\frac{r^2}{2\sigma^2} - 1\right) \cdot \frac{r}{\sigma^4} \cdot e^{-\frac{r^2}{2\sigma^2}}. \quad (2.14)$$

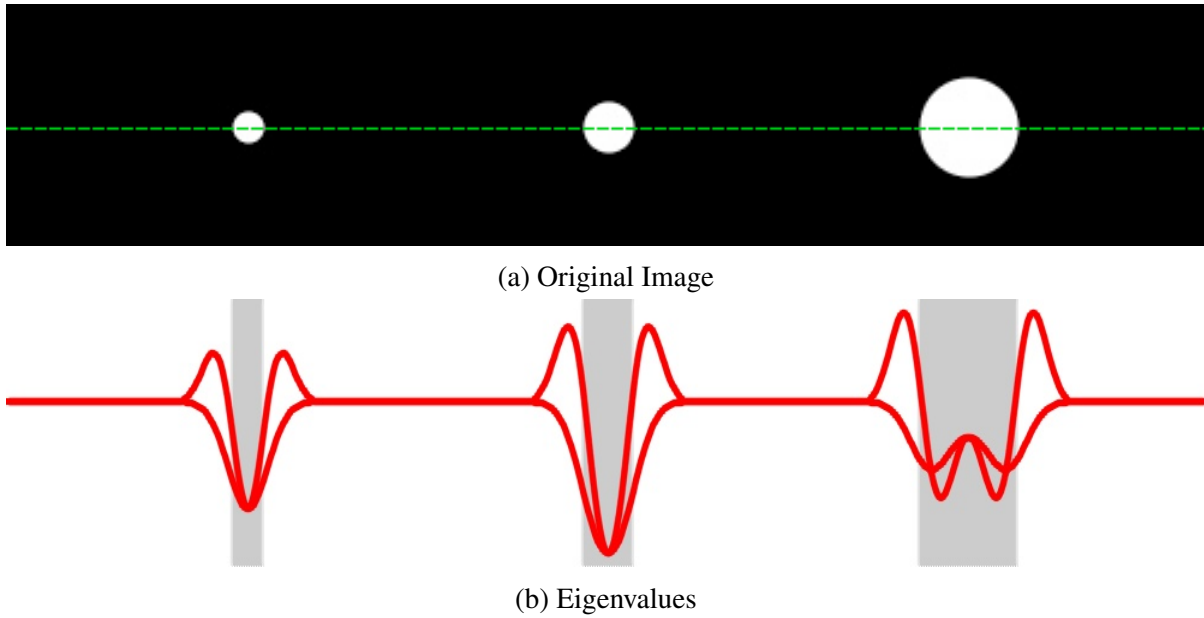


Figure 2.11: Eigenvalues of The Hessian Matrix for 2D Balls

When $r = \sqrt{2}\sigma$, the partial derivative above equals to zero and $\mathcal{R}(\sigma, r)$ reaches minimum

$$\mathcal{R}(\sigma, \sqrt{2}\sigma) = -\frac{1}{\sigma^2 e}. \quad (2.15)$$

We compute the eigenvalues of the Hessian matrix for Figure 2.9. We plot the eigenvalues along the centre row of the image as illustrated by Figure 2.11a. The sigma of the Gaussian is $\frac{10}{\sqrt{2}}$, which matches the radius of the second ball. As a result, Figure 2.11b shows that the highest response at the centre of the second ball.

Now we need to combine these two eigenvalues as one measure. The following are all reasonable options:

- $-(\lambda_1 + \lambda_2)$, Figure 2.12(a)
- $\lambda_1 \lambda_2$, Figure 2.12(b)
- $\lambda_1^2 + \lambda_2^2$, Figure 2.12(c)
- $\max(|\lambda_1|, |\lambda_2|)$, Figure 2.12(d)
- $-\min(\lambda_1, \lambda_2)$, Figure 2.12(e)

We have more discussion about the combination of eigenvalues in Section 2.5.

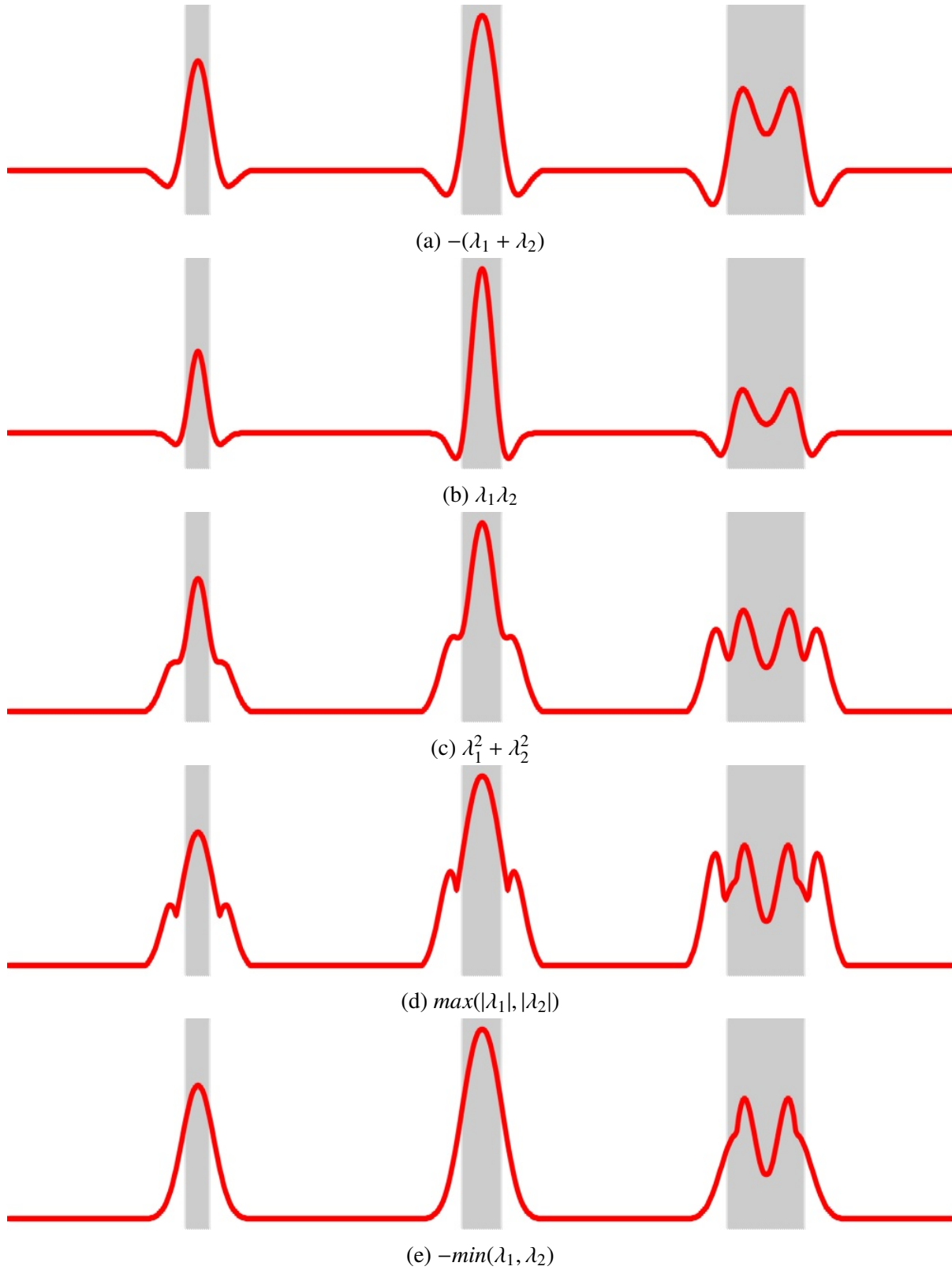


Figure 2.12: Different Ways of Combination of Eigenvalues of Balls

2.4.4 Hessian for Vessels in 3D

Hessian matrix in 3D is given by the following equation,

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f}{\partial x_3 \partial x_1} & \frac{\partial^2 f}{\partial x_3 \partial x_2} & \frac{\partial^2 f}{\partial x_3^2} \end{bmatrix}.$$

There are three eigenvalues λ_1 , λ_2 and λ_3 with the corresponding eigenvectors v_1 , v_2 and v_3 . We sort the eigenvalues so that,

$$|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$$

The Hessian matrix for 3D vessels is very similar to Hessian for 2D balls. For the three eigenvalues of the Hessian in 3D, one of them should be very close to zero because the intensity of the image stays constant along the orientation of the vessel.

$$|\lambda_1| \approx 0$$

The cross section of the 3D vessels are balls. The other two eigenvalues are equivalent to the eigenvalues of the 2D Hessian calculated from the cross section of the vessel. Therefore, at the centre of the vessel, the two eigenvalues should be approximately equal to each other and their absolute value should be much greater than zero.

$$|\lambda_2| \approx |\lambda_3| \gg 0$$

2.4.5 Hessian Matrix in General

In mathematics, the Hessian matrix is a square matrix of second-order partial derivatives of a function. In our context, function f is image which can be view as either a 2D or 3D discrete function.

$$\mathcal{H}(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}. \quad (2.16)$$

To analyze the local feature of an image, it is a common approach to consider the neighbours of an image $\mathcal{I}(x)$ at a point x using Tyler expansion [19],

$$\mathcal{I}(x + \Delta x) \approx \mathcal{I}(x) + \Delta x^T \mathcal{J}(x) + \Delta x^T \mathcal{H}(x) \Delta x \quad (2.17)$$

This approximates the image up to second order. $\mathcal{J}(x)$ is the Jacobian matrix of the image at position x , which is also equivalent to the gradient of the image $\nabla \mathcal{I}$. Δx is offset of the image position, and $x + \Delta x$ give the position of the neighbouring location. And $\mathcal{H}(x)$ is the Hessian matrix computed from the image at position x and it contains the information about the curvature of the image function.

Image \mathcal{I} is blurred from the original image \mathcal{I}_o with a Gaussian filter $\mathcal{G}(\sigma)$

$$\mathcal{I} = \mathcal{G}(\sigma) * \mathcal{I}_o$$

The Hessian matrix compute the second order derivative of the function. For each image position, the i^{th} row and j^{th} column of the Hessian matrix is,

$$\mathcal{H}_{ij}(x) = \frac{\partial}{\partial x_i} * \frac{\partial}{\partial x_j} * \mathcal{G}(x, \sigma) * \mathcal{I}(x)$$

where $\frac{\partial}{\partial x_i}$ is the derivative on the i^{th} dimension and $\mathcal{G}(x, \sigma)$ is a Gaussian centred at x . Notice that both the derivative of the image $\frac{\partial}{\partial x_i}$ and the Gaussian filter $\mathcal{G}(x, \sigma)$ can be represented by convolution of matrices.

Let λ_k denote the eigenvalue corresponding to the k^{th} normalized eigenvector u_k of the Hessian matrix $\mathcal{H}(x)$. From the definition of eigenvalues,

$$\mathcal{H}(x)u_k = \lambda_k u_k$$

Left multiply both sides of the equation gives,

$$u_k^T \mathcal{H}(x) u_k = \lambda_k.$$

The benefits of eigenvalue analysis is to that it automatically extracts the principal orientation which gives the smallest and biggest semi-axis of the corresponding ellipsoid represented by the matrix. The value of the k-th semi-axis is corresponding to $\frac{1}{\sqrt{|\lambda_k|}}$. As for the Hessian matrix, using the eigenvalue analysis, the local second order structure can be decomposed and this directly gives the direction of smallest curvature [19]. The direction of the smallest curvature is the orientation of the vessel.

2.5 Combination of Eigenvalues

Some examples of combination of eigenvalues were discussed in Section 2.4.3. A standard combination of the eigenvalues for vesselness measure [19] is introduced in Section 2.5.1. We also developed a alternation of the vesselness measure for ballness measure in Section 2.5.2.

2.5.1 3D Vesselness Measure

We sort the eigenvalues so that,

$$|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$$

If we are detecting bright vessels on dark background, both λ_2 and λ_3 should be less than zero based on the previous discussion. If any of them are grater than zero, that voxel is most likely a background voxel (vessel measure is set as zero). If both λ_2 and λ_3 are greater than zero, the following three components are used for vesselness measure [19]:

- To differentiate between plate and line like structures,

$$\mathcal{A} = \frac{|\lambda_2|}{|\lambda_3|}.$$

$\mathcal{A} \rightarrow 0$ implies a plane; $\mathcal{A} \rightarrow 1$ implies a line. This term can also be consider as the roundness of the vessel.

- To differentiate blob like structure,

$$\mathcal{B} = \frac{|\lambda_1|}{\sqrt{|\lambda_2 \lambda_3|}}.$$

$\mathcal{B} \rightarrow 1$ implies that $|\lambda_1| \approx |\lambda_2| \approx |\lambda_3|$, which implies a blob like structure with equivalent curvature along all directions. Therefore, when building a vessel detector, we looking for the opposite of \mathcal{B} .

- To differentiates between foreground (vessel) and background (noise),

$$\mathcal{S} = \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}$$

The smaller \mathcal{S} is, the more likely the voxel belongs to background. Based on the previous discussion on eigenvalues on 2D balls (Figure 2.12c), it is obvious that \mathcal{S} has the highest value when close to the centreline of the vessel.

Finally, the formulation of the vessel measure by Frangi et al. [19] is as follows,

$$\mathcal{V} = \begin{cases} 0 & \text{if } \lambda_2 > 0 \text{ or } \lambda_3 > 0 \\ (1 - e^{-\frac{\mathcal{A}^2}{2\alpha^2}})e^{-\frac{\mathcal{B}^2}{2\beta^2}}(1 - e^{-\frac{\mathcal{S}^2}{2\gamma^2}}) & \text{otherwise} \end{cases} \quad (2.18)$$

The exponential function is used in order to map the measure to a value between 0 and 1. α , β and γ are parameters to tune.

2.5.2 2D Ballness Measure

A ball structure does not have any principle direction. Therefore, the two eigenvalues should be close to each other.

We also sort the eigenvalues so that

$$|\lambda_1| \leq |\lambda_2|$$

Similarly, if we are detecting white balls on dark background, and if either λ_1 or λ_2 is smaller than zero, the ball measure is set to zero. Otherwise, the following two components are used for the ballness measure.

- To differentiate between plate and line like structures,

$$\mathcal{A} = \frac{|\lambda_1|}{|\lambda_2|}$$

$\mathcal{A} \rightarrow 0$ implies a line; $\mathcal{A} \rightarrow 1$ implies ball.

- To differentiates between foreground (ball) and background (noise),

$$\mathcal{S} = \sqrt{\lambda_1^2 + \lambda_2^2}$$

The smaller S is, the more likely the voxel belongs to background.

Finally, the ballness measure can be formulated as,

$$\mathcal{V} = \begin{cases} 0 & \text{if } \lambda_1 > 0 \text{ or } \lambda_2 > 0 \\ (1 - e^{-\frac{\mathcal{A}^2}{2a^2}})(1 - e^{-\frac{S^2}{2\gamma^2}}) & \text{otherwise} \end{cases} \quad (2.19)$$

The different terms of ballness are visualized in Figure 2.13. The sigma is equal to $5\sqrt{2}$, which matches the size of the second ball. Therefore, the second ball has the highest ballness response.

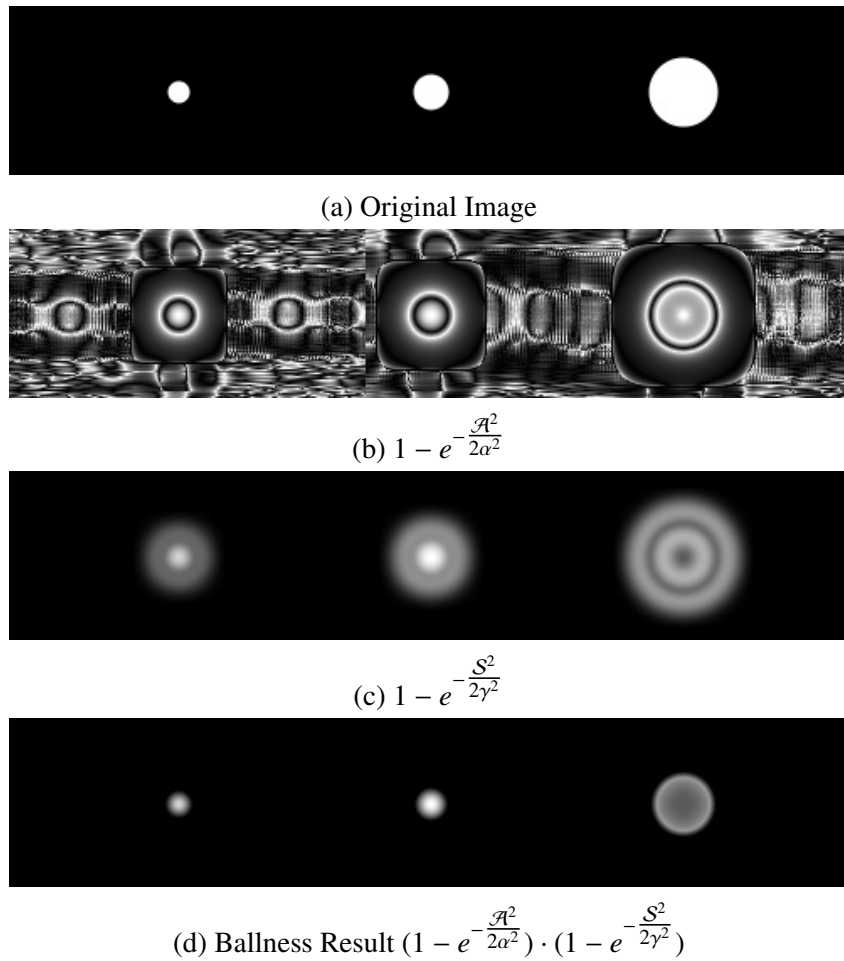


Figure 2.13: Ballness

2.6 Comparison Between Scale

Scale is always an important factor for feature detector such as the SIFT feature [30]. If we have two pictures of a same object taken from different distances, their sizes are different on the image. A good feature detector should still able to recognize them.

For vesselness measure, it is also very important to retrieve the size of the vessels. There are multiple ways to achieve this [19, 31]. We adopted the method proposed by Frangi et al.[19]. The best response is selected among all scales $\mathcal{R}(\sigma) = \max_{\sigma_i} \mathcal{R}(\sigma_i)$.

From Equation (2.10), the best response of the convolution of the second derivative in 1D with a box is,

$$\begin{aligned} \mathcal{R}(\sigma) &= \int_{-\sigma+\mu}^{\sigma+\mu} \frac{\partial^2 \mathcal{G}(x)}{\partial x^2} f(x) dx \\ &= -\frac{1}{\sigma^2} \cdot \frac{\sqrt{2}C}{\sqrt{e\pi}} \end{aligned}$$

From Equation (2.15), we know the best response of the convolution of the second derivative in 2D with a ball is,

$$\begin{aligned} \mathcal{R}(\sigma) &= \iint \frac{\partial^2 \mathcal{G}(x, y)}{\partial x^2} \cdot \mathcal{F}(x, y) dx dy \\ &= -\frac{1}{\sigma^2} \cdot \frac{C}{e} \end{aligned}$$

Therefore, to make the the convolution result invariant to scale σ , we need to normalized our Gaussian filter with a scaler σ^2 .

$$\mathcal{G}'(\sigma) = \sigma^2 \cdot \mathcal{G}(\sigma) \tag{2.20}$$

However, not every term in the vessel measure in Equation (2.18) is affected by the scale problem. This following term is affected by the problem.

$$\mathcal{S} = \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}$$

These following two are irrelevant to scale because we are computing the ratio of the eigenvalues.

$$\mathcal{A} = \frac{|\lambda_2|}{|\lambda_3|}, \quad \mathcal{B} = \frac{|\lambda_1|}{\sqrt{|\lambda_2 \lambda_3|}}$$

2.7 Result

Figure 2.14 show the vesselness with different sigmas. Notice that with a small sigma, we can detect a lot of small vessels (Figure 2.14b). When we increase σ , we start to detect bigger vessels however we lose the small ones (Figure 2.14c-d).

The vesselness is computed for all different scales and we choose the scale with the maximum response. A comparison of the results of original image and the vesselness result are shown in Figure 2.15 and Figure 2.16 with the visualization method discussed in Appendix A.1. Figure 2.15b shows the result with *maximum intensity projection*. Figure 2.15c shows the orientation of the vessels. Figure 2.16 show some arbitrary cross sections of the 3D volume of the original data and vesselness filter. Notice that there is a grey background in the original data, while the background noise is suppressed to a great extent in the vesselness measure.

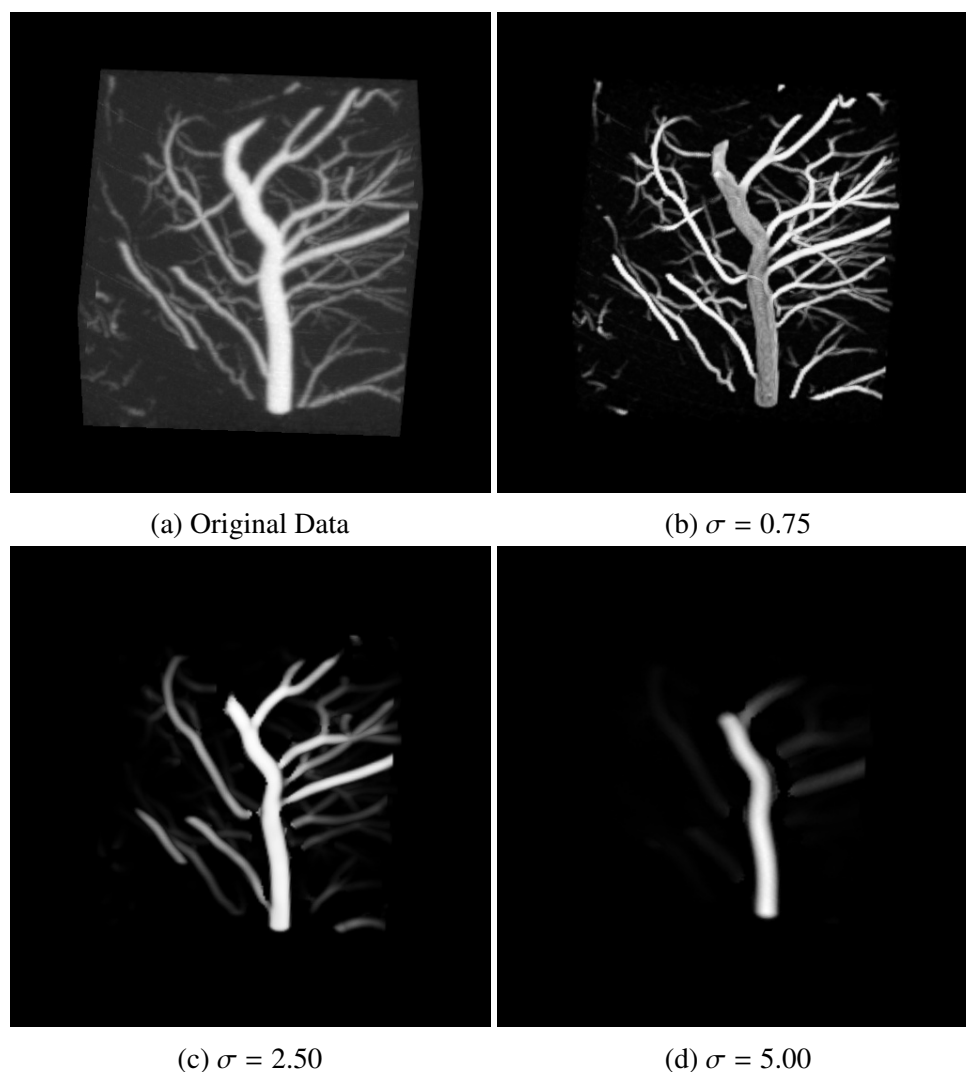
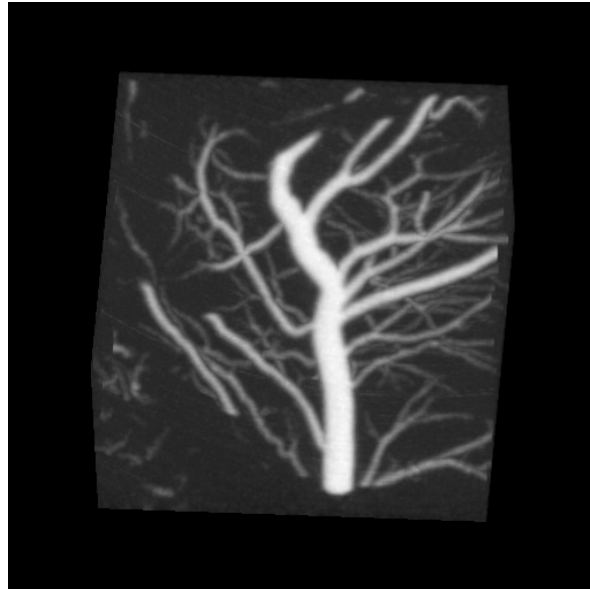
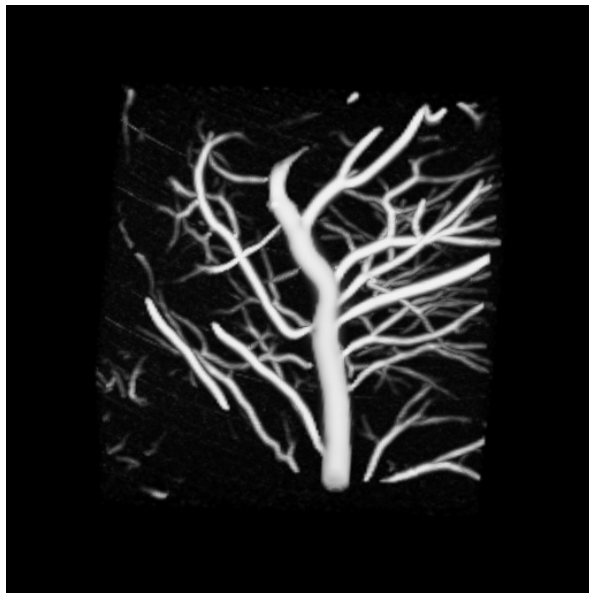


Figure 2.14: Vesselness With Different Sigmas



(a) Original Data



(b) Vesselness



(c) Vessel Direction

Figure 2.15: Comparing Original Data and Vesselness

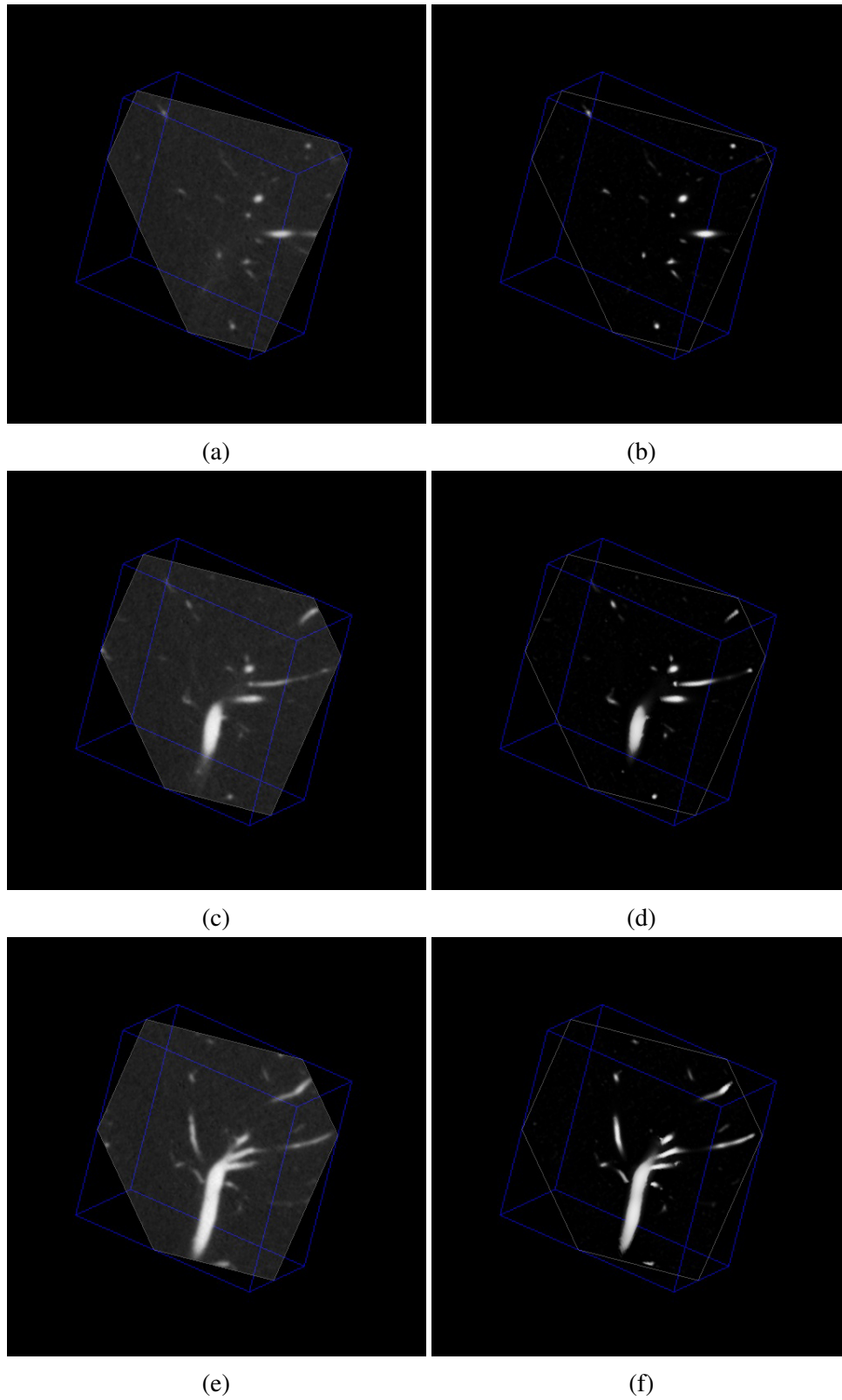


Figure 2.16: Comparing Original Data and Vesselness (Cross Sections)

Chapter 3

Centreline Extraction

3.1 Overview

The formal definition of centreline is given in Section 1.1.2. There are a couple of motivations for extracting the centreline of the vessels:

1. in the context of information theory, we can make the data sparse so that we need less number of bits to encrypt the data;
2. we can compute the topology of the vessels using *minimum spinning tree*;
3. it is straight forward to reconstruct the segmentation of the vessel given the correct centreline.

Some related methods about extracting centreline are resented in Section 1.2.4. We develop two centreline approaches in this chapter: (1) Vessel Thinning; (2) Model Fitting.

Vessel Thinning

We get some intuition from Canny edge detector [8], which was developed by John F. Canny in 1986. Canny edge detector has been one of the most commonly used edge detectors in image processing, detecting edges in a very robust manner. The algorithm contains multiple steps:

1. noise reduction using the Gaussian filter;
2. finding the intensity gradient of the image;
3. *non-maximum suppression* (keeps only the pixels on an edge with the highest gradient magnitude and suppress the others);
4. tracing edges through image with *hysteresis thresholding*.

According to the previous discussion in Chapter 2, vesselness measure is the highest at the centreline of the vessels. If we apply *non-maximum suppression* to the vesselness measure, we should be able to extract the centreline of the vessel. This will be further explained in Section 3.2.

Model Fitting

There are many possible geometrical models that may fit to our data: 1) lines; 1) line intervals; 2) cylinders; 3) balls; 4) points with orientation. The cylinder model sound most reasonable for our data because the vessels can be intuitively viewed as a set of tubes. Section 3.3 briefly introduce model fitting method. This part of work is carried out by a colleague, Xuefeng Chang. Please refer to him for more detail.

3.2 Vessel Thinning

The vessel thinning is inspired by Canny edge detector [8]. We adopt two important steps in Canny edge detector to vessel centreline detection. That is *non-maximum suppression* and *hysteresis thresholding*. We also found the similar idea in [42].

3.2.1 Non-maximum Suppression

Non-maximum suppression is a critical step in Canny edge detector for edge thinning. A search is carried out along the gradient direction of the image. If the magnitude of a pixel is smaller than the magnitude of any of its two neighbours in the gradient direction, it will be suppressed (by setting its value to 0).

Similarly, for vesselness measure, we suppress a point if it's value is smaller than any of the neighbours in the normal direction of the vessel orientation. For implementation in 2D, we will categorize the vessel orientation into one of the following four major directions¹:

- 0 or 180 degrees: a pixel is suppressed if its magnitude is less than the pixel that is above or the pixel that is below;
- 45 or 225 degrees: a pixel is suppressed if its magnitude is less than the pixels that is top left corner or the one on the bottom right corner;
- 90 or 270 degrees: a pixel is suppressed if its magnitude is less than the pixel that is on the left or the pixel on the right;

¹Notice that the sign of the orientation is irrelevant, therefore 0 degree is equivalent to 180 degrees

Orientation	Comparing Neighbours
(0, 0, 1)	(0, 1, 0), (1,-1, 0), (1, 0, 0), (1, 1, 0), (-1,-1, 0), (-1, 0, 0),(-1, 1, 0), (0,-1, 0)
(0, 1,-1)	(0, 1, 1), (1,-1,-1), (1, 0, 0), (1, 1, 1), (-1,-1,-1), (-1, 0, 0),(-1, 1, 1), (0,-1,-1)
(0, 1, 0)	(0, 0, 1), (1, 0,-1), (1, 0, 0), (1, 0, 1), (-1, 0,-1), (-1, 0, 0),(-1, 0, 1), (0, 0,-1)
(0, 1, 1)	(0, 1,-1), (1,-1, 1), (1, 0, 0), (1, 1,-1), (-1,-1, 1), (-1, 0, 0),(-1, 1,-1), (0,-1, 1)
(1,-1,-1)	(0, 1,-1), (1, 0, 1), (1, 1, 0), (-1,-1, 0), (-1, 0,-1), (0,-1, 1)
(1,-1, 0)	(0, 0, 1), (1, 1,-1), (1, 1, 0), (1, 1, 1), (-1,-1,-1), (-1,-1, 0),(-1,-1, 1), (0, 0,-1)
(1,-1, 1)	(0, 1, 1), (1, 0,-1), (1, 1, 0), (-1,-1, 0), (-1, 0, 1), (0,-1,-1)
(1, 0,-1)	(0, 1, 0), (1,-1, 1), (1, 0, 1), (1, 1, 1), (-1,-1,-1), (-1, 0,-1),(-1, 1,-1), (0,-1, 0)
(1, 0, 0)	(0, 0, 1), (0, 1,-1), (0, 1, 0), (0, 1, 1), (0,-1,-1), (0,-1, 0), (0,-1, 1), (0, 0,-1)
(1, 0, 1)	(0, 1, 0), (1,-1,-1), (1, 0,-1), (1, 1,-1), (-1,-1, 1), (-1, 0, 1),(-1, 1, 1), (0,-1, 0)
(1, 1,-1)	(0, 1, 1), (1,-1, 0), (1, 0, 1), (-1, 0,-1), (-1, 1, 0), (0,-1,-1)
(1, 1, 0)	(0, 0, 1), (1,-1,-1), (1,-1, 0), (1,-1, 1), (-1, 1,-1), (-1, 1, 0),(-1, 1, 1), (0, 0,-1)
(1, 1, 1)	(0, 1,-1), (1,-1, 0), (1, 0,-1), (-1, 0, 1), (-1, 1, 0), (0,-1, 1)

Table 3.1: Non-maximum Suppression in 3D

- 135 or 315 degrees: a pixel is suppressed if its magnitude is less than the pixel that is on the top right corner or the one on the bottom left corner.

In 3D, we have a 26-neighbourhood system instead of having a 8-neighbourhood system. There are 13 major orientation instead of 4 in 2D. Instead of comparing the magnitude of the pixel with the neighbouring pixels that on the perpendicular direction, we need to compare the magnitude of a voxel with the neighbouring voxels that are on the cross section. The number of voxels on the cross section may be either 8 voxels or 6 voxels. In Table 3.1, the first column shows the 13 major orientations and the second column shows the corresponding neighbouring voxels that need to compare. There are three situations for the major orientations, as illustrated in Figure 3.1. The blue arrow is the major orientation of the vessel and the boxes are the neighbouring voxels that we need to compare with the centre voxel. In Figure 3.1a, the major orientation of the vessel is perpendicular to two axes. In Figure 3.1b, the major orientation is perpendicular to one axis. In Figure 3.1c, the major orientation is perpendicular to none of the axes.

3.2.2 Hysteresis Thresholding

Non-maximum suppression (Section 3.2.1) is able to thin the data to a large extend. However, the magnitude of the background is extremely unpredictable, which results in a lot of noisy points.

Since the magnitude of the background is generally lower than the vessels, we may threshold the data to clean it up. The problem is that if we use a high threshold, we may lose some

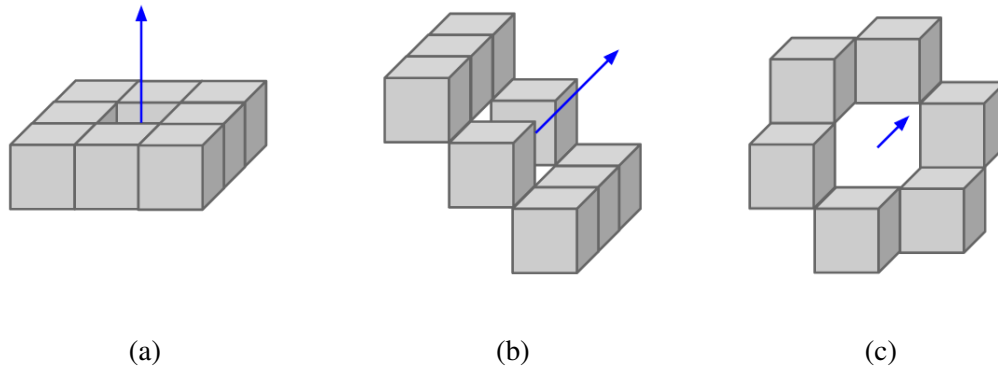


Figure 3.1: Different Situations When Comparing Neighbours With Non-maximum Suppression in 3D

fine details or small branches of the vessels; if we use a low threshold, we may pick up some undesired noise. This is the major motivation where the hysteresis thresholding in Canny edge detector comes from.

We use two thresholds - high and low. Using the first threshold (high), we mark down the voxels that we can be fairly sure belong to the skeleton. And we use those voxels as our seed points and do region growing from them until we reach the second threshold (low).

Once this process is complete we have a binary image where each voxel is marked as either skeleton voxel or non-skeleton voxel. And we can get a set of vessel branches with fairly good connectivity.

3.3 Model Fitting

When doing model fitting, we describe the vessel as a set of line intervals with radii instead of data points as described in Section 3.2. Most of the work about model fitting is done by a colleague, Xuefeng Chang. In this thesis, we only presents some fundamental theories about model fitting and a toy version of ball fitting algorithm on 2D synthetic data.

The reason for ball fitting is as follows. Xuefeng fit line intervals into the thresholded vesselness data. For each of the line intervals that is fitted into the data, there is a parameter σ which describes the distribution of the data points around the line interval. This is related to the scale of a vessel. Remember that in Chapter 2, there is also a σ that is related to the scale of a vessel. Both of the sigmas have the same geometrical meaning, but they are not the same. We make use of the image intensity when doing ball fitting. We want to do the ball fitting without computing a ‘ballness measure’. Eventually, we want to do cylinder fitting based image intensity for 3D data.

The model fitting problem can be formulated as an energy minimization problem. The energy function proposed by Delong et al. [12] is justified by information theory. And it fits well in our problem. The energy function is [12]:

$$E(f; \theta) = \sum_{p \in \mathcal{P}} D_p(f_p, \theta_{f_p}) + \sum_{p, q \in \mathcal{N}} V_{p, q}(f_p, f_q) + \sum_{l \subseteq \mathcal{L}} h_l(\theta) \cdot \delta_l(f) \quad (3.1)$$

where $D_p(f_p, \theta_{f_p})$ is the data cost for assigning a pixel p to a label f_p ; $V_{p, q}(f_p, f_q)$ is the smooth cost for two pixels p and q in a neighbourhood system \mathcal{N} ; and $h_l(\theta) \cdot \delta_l(f)$ is the label cost for using label f .

3.3.1 Line Interval Fitting with PEARL framework

The line interval fitting is developed under PEARL framework proposed by Hossam and Boykov [21]. There are three critical steps under this framework, namely: 1) Propose; 2) Expand; 3) Re-estimate Labels.

Propose

The algorithm begins with some randomly sampled initial models. In this case, the models are line intervals. Each line interval has 7 parameter: two coordinates in 3D for the end points of the lines interval and a sigma which describes the distribution of the data point. We also refer sigma as the thickness of the line interval.

Expand

Run α -expansion [6] to assign data points to models. The interval is modelled as a mixture of Gaussians $\mathcal{N}(\mu, \sigma^2)$ for each μ interpolating a and b [12]. We use the *log* likelihood of the following Equation (3.2) as the data cost for the graph cuts energy Equation (3.1).

$$Pr(x|a, b, \sigma^2) = \int_0^1 \mathcal{N}(x|(1-t)a + tb, \sigma^2) \quad (3.2)$$

We use the pair-wise smooth term proposed by [34] as the smooth cost of the graph cuts energy Equation (1.1). A pair-wise smooth term is illustrated in Figure 3.2. \tilde{p} and \tilde{q} are two data points which are assigned to two line intervals l_1 and l_2 respectively. Take \tilde{q} as an example, we first project \tilde{q} to l_2 and the projection point is q . We assume that q is the real position of \tilde{q} . And then we project point q to l_1 and get the projection point q' . Similarly, we can have p and p' through two projections for point \tilde{p} . Finally, we use the following as the pair-wise smooth

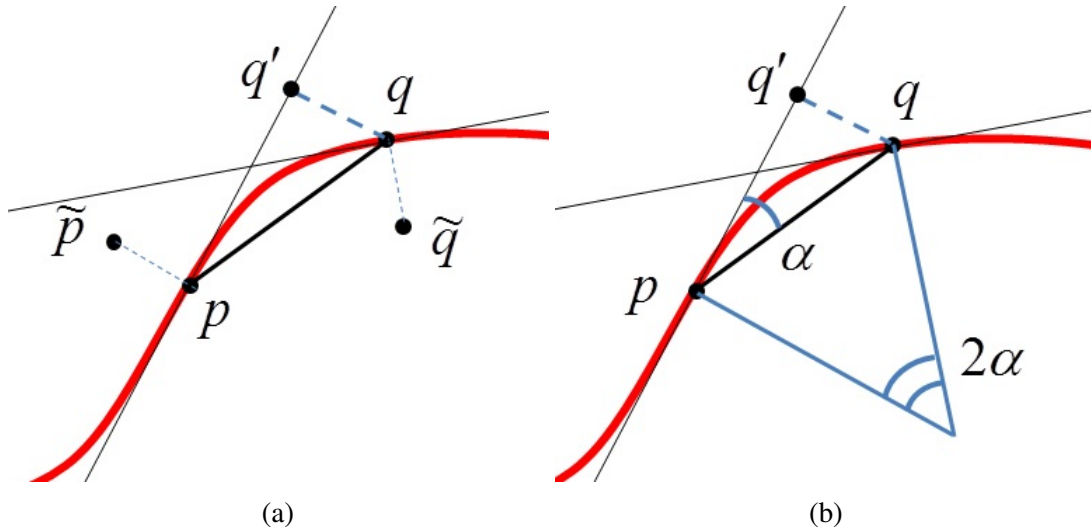


Figure 3.2: Pairwise Interaction Approximating Curvature [34]

cost.

$$\frac{|p - p'| + |q - q'|}{|p - q|^2}, \quad (3.3)$$

As is shown in Figure 3.2b, the quotient $\frac{|q - q'|}{|p - q|^2}$ yields half the curvature at p under the assumption that p and q belong to a constant curvature segment [34].

Re-estimate Labels

Finally, we need to re-estimate the models based on the data points that are assigned to the model. Any optimization methods that can minimize the energy function mentions above (in Subsection **Expand**) can be used. Since the model is too complicated, we currently use exhaustive search for Re-estimation.

3.3.2 Ball Fitting

When using the Hessian matrix in Section 2.4 to detect vesselness, there is a sigma (σ_1) which tells us a the size of the vessel at that point. According to Equation (2.14), the relationship between σ_1 and the radius of the vessel is: $r = \sqrt{2}\sigma_1$. When doing model fitting, there is another sigma (σ_2), which describes the distribution of the data. This sigma (σ_2) is also related to the size of the vessel. These two sigmas are so similar to each other, but, not the same. We are exploring a way to merge these two sigmas together. In another words, we are seeking for a way to do line segment fitting and vesselness filtering instead of doing doing them in two separate phrases.

According to Section 2.3.2, we believe that if can figure a way to merge ballness measure

(Section 2.5.2) and ball fitting for 2D images, we are able to solve the similar problem for vessels in 3D. That's the reason why we are doing ball fitting in this section. But we haven't found a way to merge ball fitting and ballness measure yet.

Propose

As in Section 3.3.1, initial models are randomly sampled.

Expand

The log likelihood of a Gaussian function gives,

$$\begin{aligned} -\log N(x|\mu, \sigma) &= -\log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) + \frac{(x - \mu)^2}{2\sigma^2} \\ &= \frac{(x - \mu)^2}{2\sigma^2} + \log(\sigma) + C, \end{aligned} \quad (3.4)$$

where C is a constant,

$$C = \frac{1}{2}\log(2\pi). \quad (3.5)$$

If we consider the ball as a Gaussian model $N(x|\mu, \sigma)$, where μ is the centre of the ball, and σ is the radius. The error function for a Gaussian can be separated into two terms,

$$E_r(\mu, \sigma) = \sum_{x \in \mathcal{S}} -\log N(x|\mu, \sigma) + \sum_{x \in \mathcal{S}} (I_o - I_x)^2 \quad (3.6)$$

where \mathcal{S} is the set of points that have been assigned to this model (or label). According to Equation (3.4), the first term, $\log N(x|\mu, \sigma)$, measure the distance square error between a data point x and the model. The second term, $(I_o - I_x)^2$, where I_o is the intensity of the ball and I_x is the intensity of the image at point x , measure the error of the intensity. For the data cost in graph cuts, we use Equation (3.6).

We still use Potts model [6] for the smooth cost. If the neighbouring pixels are labelled as the same model, the smooth cost is zero; otherwise, the smooth cost is set to a constant value.

Re-estimation

During re-estimation, we need to determine a model which fits the data better. We can achieve this by taking the partial derivative of the error function Equation (3.6).

We first fit the centre μ by taking partial derivative of the error function over μ .

$$\frac{\partial E_r(\mu, \sigma)}{\partial \mu} = \sum_{x \in S} \frac{(x - \mu)}{\sigma^2} = 0. \quad (3.7)$$

The above yields

$$\mu = \frac{1}{|S|} \sum_{x \in S} x, \quad (3.8)$$

which indicates that the new center should be the center of mass of the data. Similarly, we take the partial derivative of the error function with respect to σ :

$$\frac{\partial E_r(\mu, \sigma)}{\partial \sigma} = \sum_{x \in S} -\frac{(x - \mu)^2}{\sigma^3} + \frac{1}{\sigma} = 0. \quad (3.9)$$

And solving the equation above yields:

$$\sigma = \sqrt{\frac{\sum (x - \mu)^2}{|S|}}. \quad (3.10)$$

3.4 Result

Figure 3.3 shows the result of data thinning. Figure 3.3a show the vesselness result we derived in Section 2. Figure 3.3b shows the result after non-maximum suppression. Notice that a large portion of the data points are suppressed. Figure 3.3c and Figure 3.3d show the result after two hysteresis thresholds respectively. After the first threshold, only some of the major points are picked up and a lot of small details are lost. After region growing with the second threshold, the result is still clean enough and the details of the vessels are picked up.

Figure 3.4 shows the result of line interval fitting. Figure 3.4b shows the result after a colleague, Xuefeng Chang, thresholded the vesselness measure. Figure 3.4c are the line intervals that fit into the data eventually. Figure 3.4d visualized the line intervals with thickness.

Figure 3.5 show the result of ball fitting on a image with 6 balls with various radii. 50 percent of random noise is added to the original image in Figure 3.5a. Figure 3.5a shows the original image with 50% of random noise. Figure 3.5b shows the initial sampled labels. Figure 3.5c shows the labelling after the first iteration. Notice that some of the pixels are still mislabelled. Figure 3.5d show the corresponding labels for Figure 3.5c. Figure 3.5e show the final labelling of the pixels and Figure 3.5f shows the corresponding labels. They describe the original image Figure 3.5a very well.

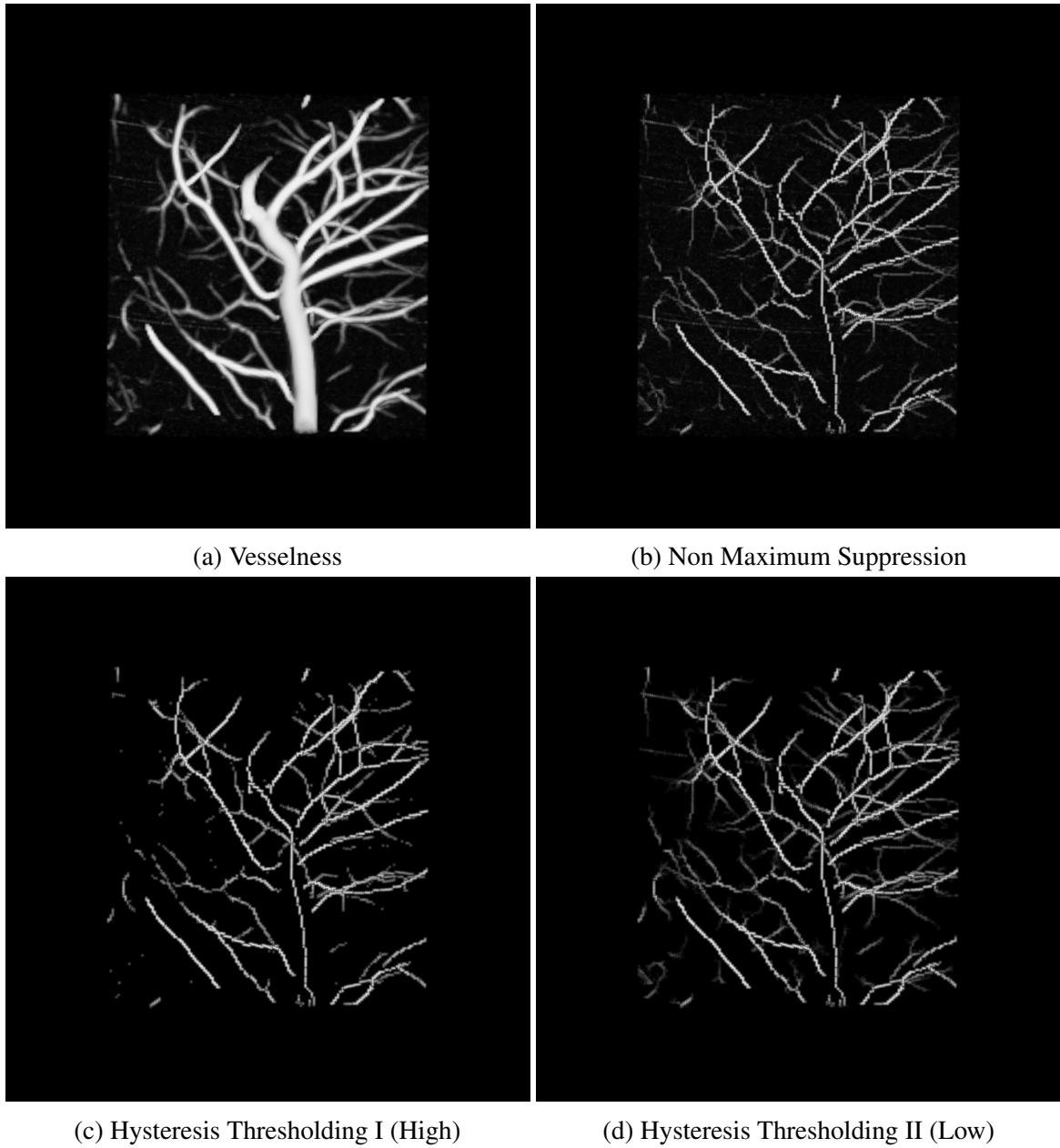


Figure 3.3: Vessel Thinning

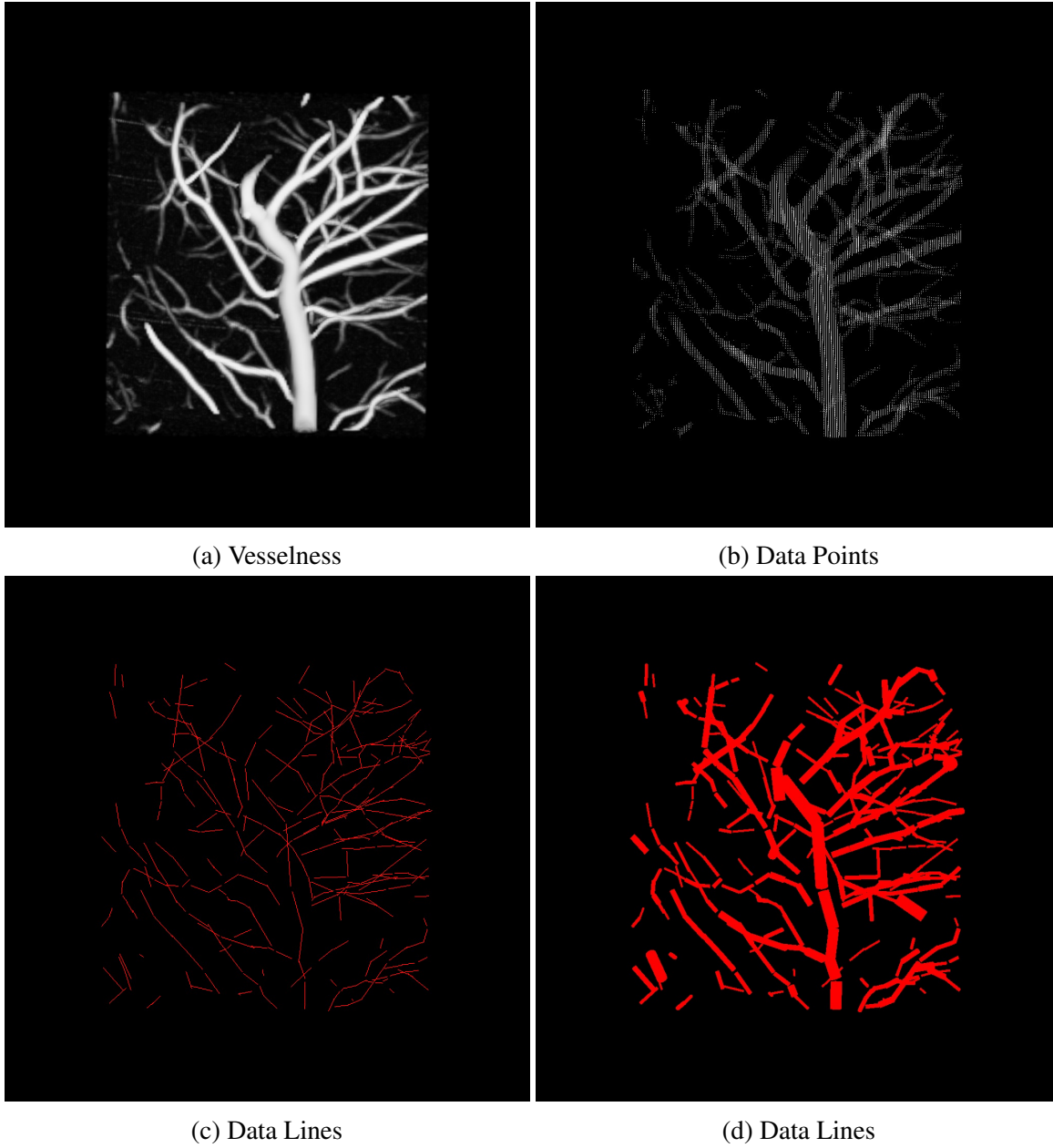


Figure 3.4: Line Interval Fitting

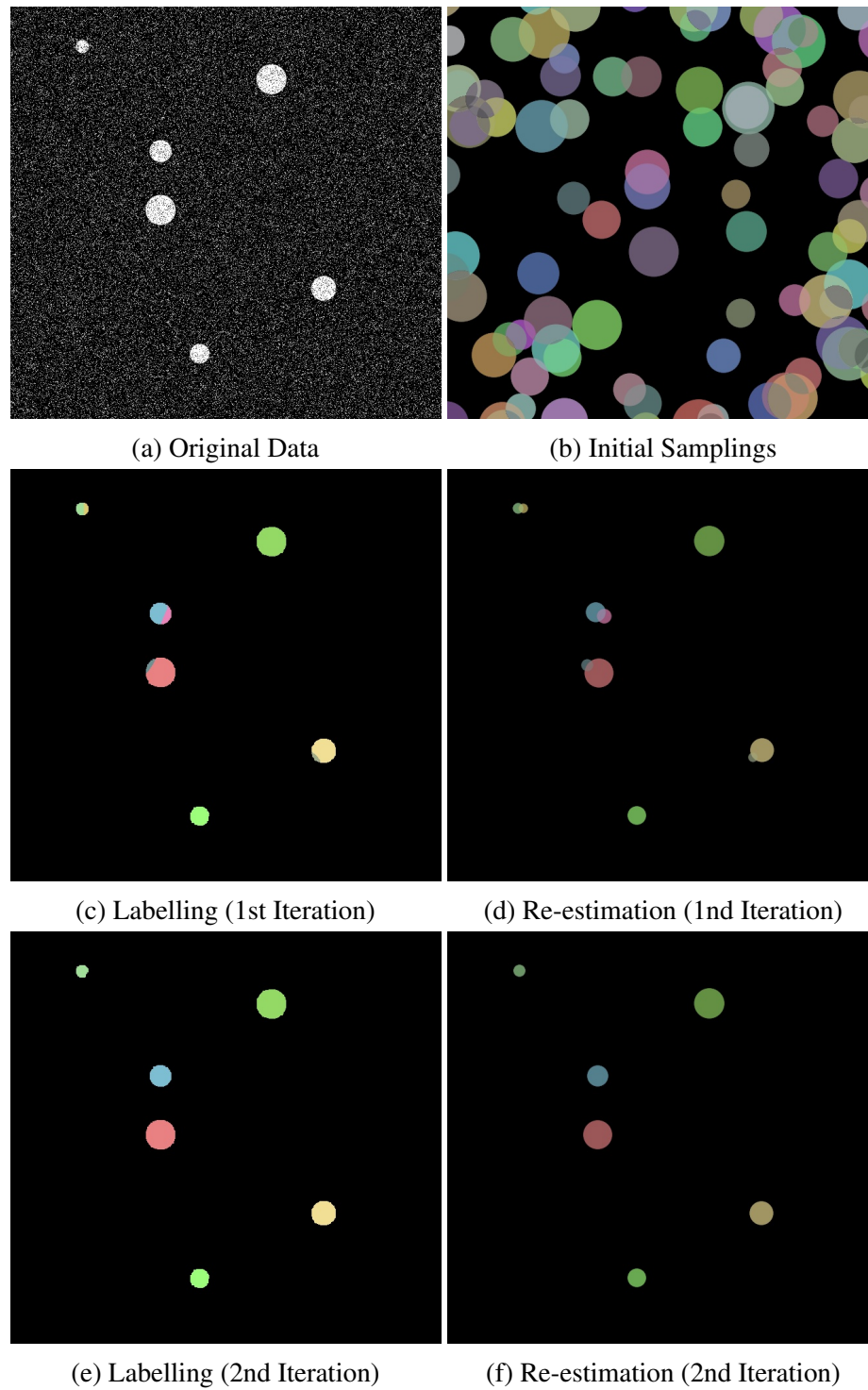


Figure 3.5: Ball Fitting

Chapter 4

Minimum Spanning Tree

A *spanning tree* is a subgraph of a connected, undirected graph that connect all vertexes. Just as the name implies, a *minimum spanning tree* is such a spanning tree that the sum of the edge weights is minimal. Figure 4.1 gives an example of a minimum spanning tree¹.

Two commonly used minimum spanning tree algorithm — Prim's algorithm and Kruskal's algorithm — are presented in Section 4.1.

Vesselness measure is weaker at bifurcations because bifurcations do not have the tubular structures. Sometimes vessel centrelines are broken down into small branches after data thinning in Section 3.2. Minimum spanning tree are used to connect these points. This is further discussed in Section 4.2.

The line intervals we get from model fitting can hardly be seamless. Section 4.3 explains how we use minimum spanning tree on lines intervals.

4.1 Prim's Algorithm and Kruskal's Algorithm

Prim's algorithm and Kruskal's algorithm are the most commonly used algorithms for finding minimum spanning tree on a connected graph.

Prim's Algorithm

- Step 1: Choose any starting vertex. Look at all edges connecting to the vertex and choose one with the lowest weight and add this to the tree.
- Step 2: Look at all edges connected to the tree. Choose the one with the lowest weight and add to the tree.
- Step 3: Repeat step 2 until all vertices are in the tree.

¹Graph is from Wikipedia

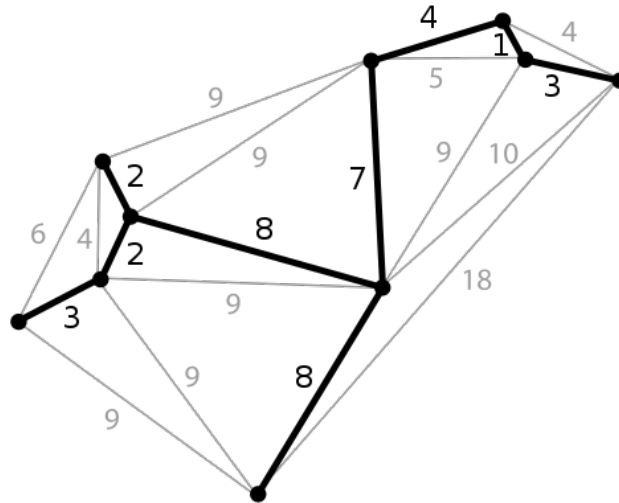


Figure 4.1: Minimum Spanning Tree on Connected Graph

Kruskal's Algorithm

- Step 1: Select an edge in order of smallest weight if it does not cause a cycle
- Step 2: Repeat step 1 until no more edges can be added

Both of them are greedy algorithms. On an original graph such as Figure 4.2. The processes of computing the minimum spanning tree using Prim's algorithm and Kruskal's algorithm are show in Figure 4.3 and Figure 4.4 respectively. We use Kruskal's algorithm in this thesis.

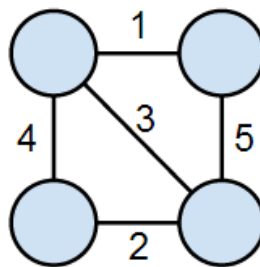


Figure 4.2: Original Graph

4.2 Minimum Spanning Tree for Points

Noticed that we have gaps in our thinned data, the most straight forward idea is to use the *morphological operator* called *Closing*. *Closing* is a combination of *Dilation* and *Erosion* in

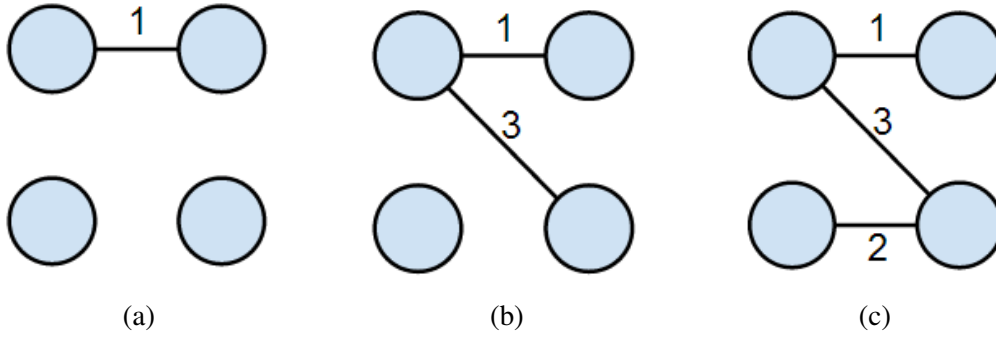


Figure 4.3: Prim's Algorithm

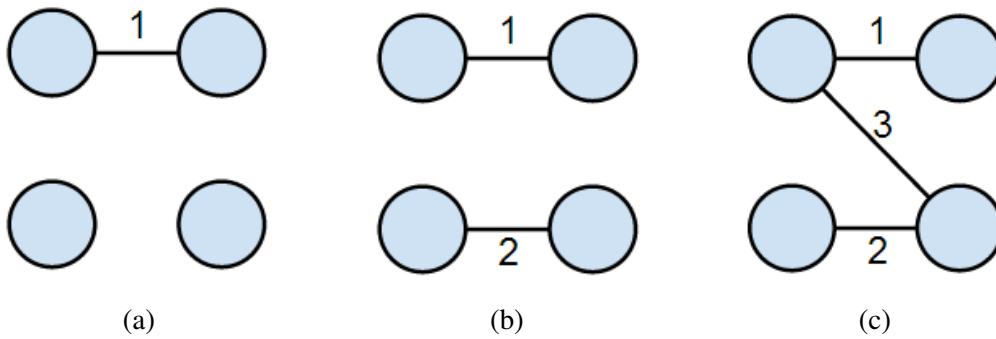


Figure 4.4: Kruskal's Algorithm

sequence. This idea failed because the *Closing* will not only close the gap between branches, it will also close any other spaces between nearby vessels.

We are only concerned about the voxels that have been left after data thinning (Section 3.2). We do not compute the distances between any two voxels and solve the minimum spanning tree problem on such a dense graph. Notice there is some connectivity in our data using 26-neighbourhood system. For example, in Figure 4.5a, there are 3 connected components as indicated by the red lines in Figure 4.5b. We refer to connected components as branches here. We can compute the distance from each branch to other branches and solve the minimum spanning tree problem on this graph. The distance between branches is defined as the following,

$$Dist(B1, B2) = \min_{x_i \in B1, x_j \in B2} Dist(x_i, x_j) \quad (4.1)$$

where $B1$ and $B2$ can be any branches, x_i and x_j are any two arbitrary points in $B1$ and $B2$ respectively. That is, the distance between two branches $B1$ and $B2$ is the minimal distance between any two points on the two branches respectively. Instead of brute force searching all combinations of voxels, we use a approximate algorithm described as follow:

- **Critical Points Detection**

Critical points are the end points of the connected components as is illustrated in Figure 4.5c. We detect these end point with *breath first search* algorithm.

- Step 1: Start with pushing a point of a branch into a queue structure;
- Step 2: Dequeue a point and push all its connected neighbours into the queue; a critical point is found if the point do not have any neighbours.

Notice that the start point of the *breath first search* may also be a critical point; therefore, we need to run the *breath first search* algorithm twice with different starting points in order to detect all critical points for a branch.

- **Breath First Search From Critical Points**

For each critical point, we run *breath first search* again until we find a point from a different branch. We add an edge to our graph: from this current critical point to the point from a different branch. Finally, we run minimum spanning tree algorithm on the graph we construct this way and get the result in Figure 4.5d.

4.3 Minimum Spanning Tree for Lines

Minimum spanning tree algorithm is a very well-defined algorithm. The only different between this section and the previous section (Section 4.2) is the construction of graph. In this case, we have line intervals as is shown in Figure 4.6a and we are looking for a minimum spanning tree such as Figure 4.6b.

Each line interval is corresponding to a node in the graph for minimum spanning tree. The weight for the graph consist of the following two parts:

$$Dist(l_1, l_2) - max(\sigma_1, \sigma_2)$$

where $Dist(l_1, l_2)$ is the shortest distance between l_1 and l_2 . σ_1 and σ_2 are the radius of the line intervals we derived during line interval fitting (Section 3.3.1). Details about computation of distance between 3d lines are available in Appendix D.

4.4 Results

Figure 4.7 show results of minimum spanning tree for thinned data. Figure 4.7a show the vesselness measure after non-maximum suppression. Figure 4.7b shows the minimum spanning

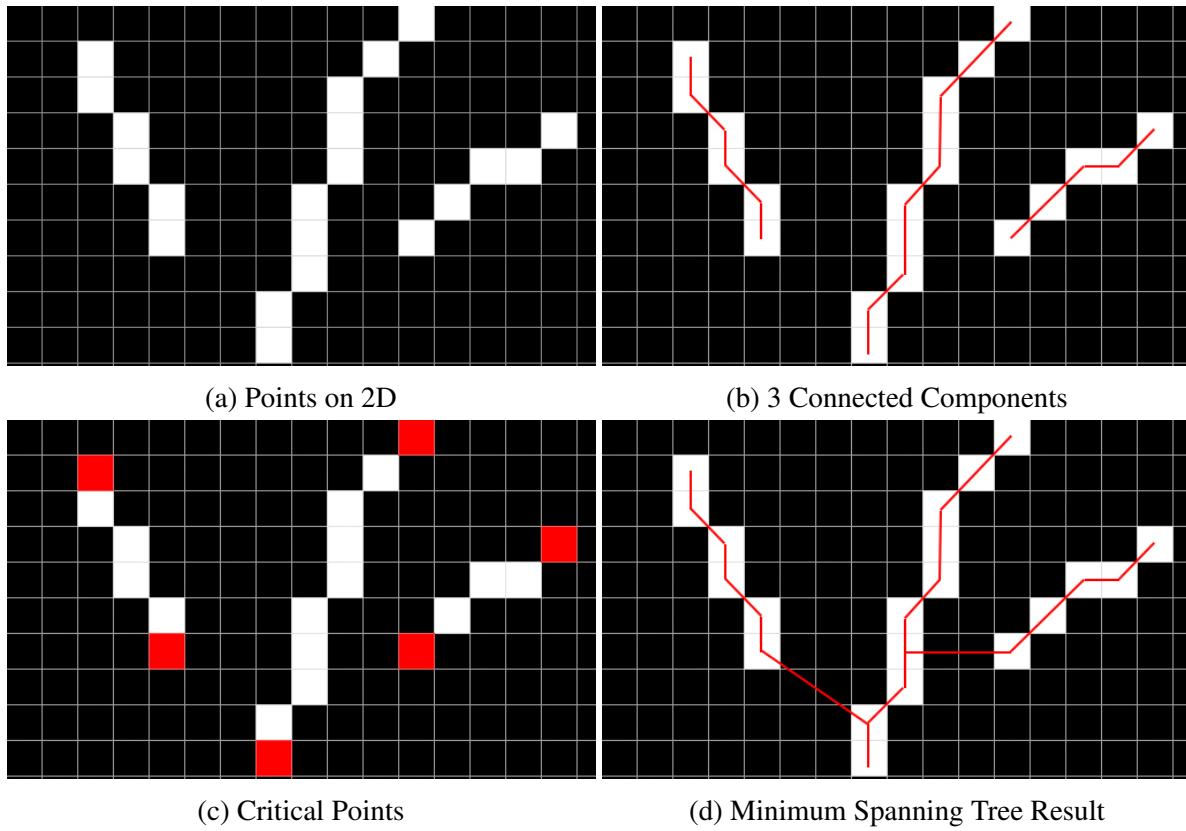


Figure 4.5: Minimum Spanning Tree on Discrete 2D Points

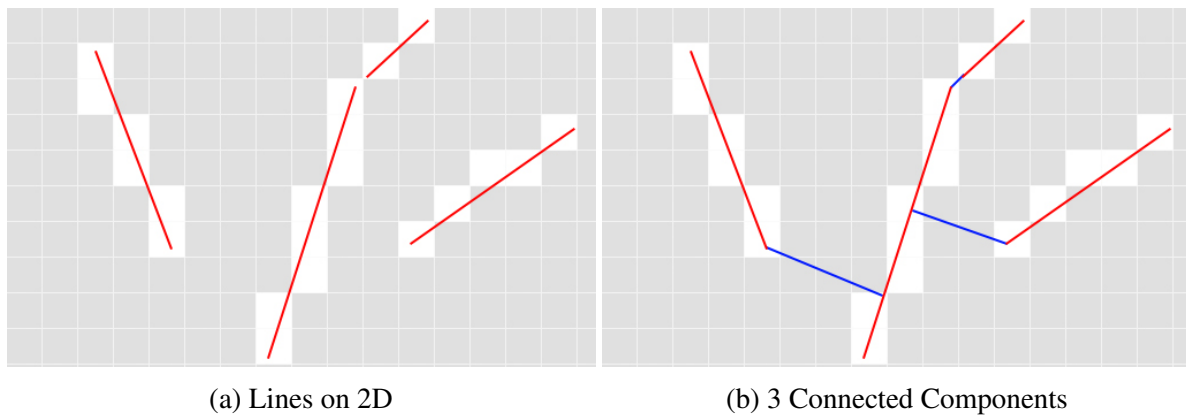


Figure 4.6: Minimum Spanning Tree on 2D Lines

tree.

Figure 4.8 shows result of minimum spanning tree for line intervals. Figure 4.8a shows the intervals that fit to the data. Figure 4.8a shows the minimum spanning tree where blue lines are the connections.

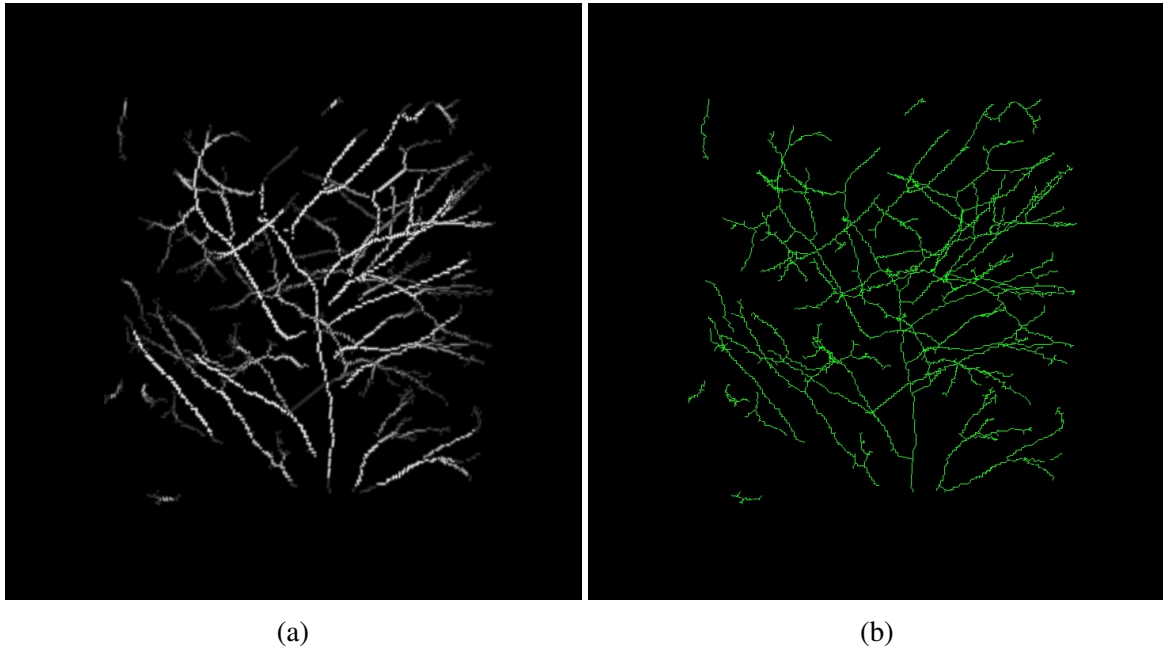


Figure 4.7: Minimum Spanning Tree on Data Thinning

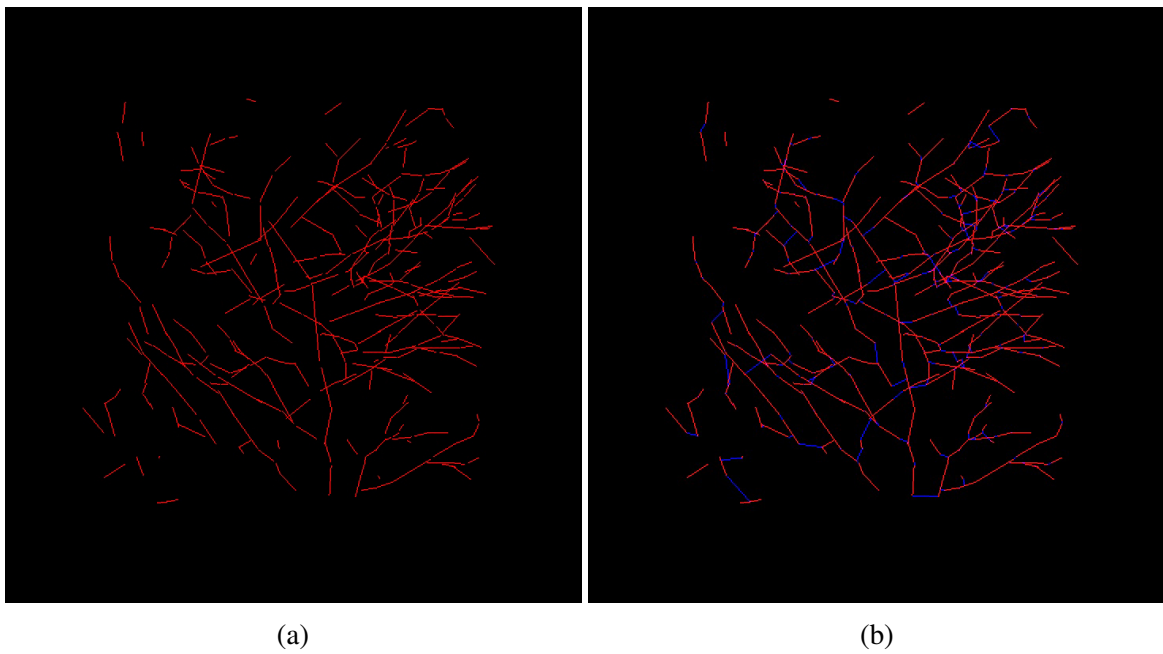


Figure 4.8: Minimum Spanning Tree on Line Intervals

Chapter 5

Conclusion

5.1 Pipeline of The Algorithm

A overall pipeline of the algorithms introduced in this thesis is shown in Figure 5.1. We first apply a rings filter in order to reduce rings artifact. Vesselness filter is used to deal with random noise. The likelihood of a voxel being vessel is generated from the vesselness filter. Orientation of the vessels is retrieved through eigenvalue decomposition. After we have the vesselness measure, two methods are used to extract the centreline of the vessels. Inspired by Canny edge detector, vessel thinning can generate a map of whether a voxel is at the centreline of the vessels or not. The other approach, line fitting, is carried out by a Colleague, Xuefeng. I have a preliminary experiment on ball fitting, which will lead us to a better model fitting in 3D in the future. Finally, a tree structure is enforced in both of the two types of centrelines that we extract. The result of vessel thinning and model fitting is very similar. The centrelines of vessel thinning is less accurate than the centrelines of model fitting. This is because vessel thinning is a local approach and the highest resolution that it can achieve is one pixel. But line intervals are fit to the data more precisely. The problem of the current model fitting is speed. It takes up to 6 hours to fit line intervals into data while vessel thinning takes less than 10 minutes.

With all these blocks in the current pipeline, we are able to extract the vessel structure as a tree-connected graph. The final result is shown in Figure 4.8b and Figure 4.7b.

5.2 Future Work

In Section 3.3.1, we fit line intervals to the data in order to extract the centreline of the vessel. The problem with this is the Gaussian mixture model is too complicated and we don't have a proper optimization algorithm. We brute forced the solution and the running speed is too slow

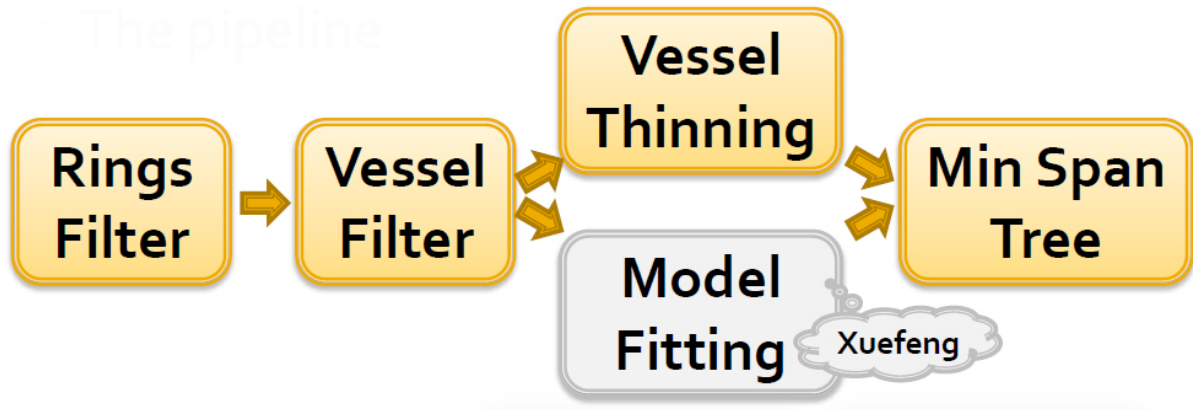


Figure 5.1: Pipeline of The Algorithms

for real life applications. Alternatively, we should try some gradient descent approach. We can also try simplify the model to line fitting instead of line interval fitting.

In Section 3.3.2, we tested the idea of ball fitting on 2D images. The motivation of doing ball fitting is to find a way to combine vesselness measure with model fitting. We will keep exploring the possibility for doing so.

Rings artifact is very common in medical images. We are currently using the rings reduction algorithm introduced by Sijibers et al. [40]. This approach has a couple of drawbacks. For example, it does not remove those rings that are close to the centre of the rings. And also, if we have partial rings, this algorithm will fail as well. We are intended to try ring fitting in colour space. We conjecture that rings fitting in colour space can at least address the partial rings problem.

Bibliography

- [1] Emran Mohammad Abu Anas, Jae Gon Kim, Soo Yeol Lee, and Md Kamrul Hasan. High-quality 3d correction of ring and radiant artifacts in flat panel detector-based cone beam volume ct imaging. *Physics in Medicine and Biology*, 56(19):6495, 2011.
- [2] Maria Axelsson, Stina Svensson, and Gunilla Borgefors. Reduction of ring artifacts in high resolution x-ray microtomography images. In *Pattern Recognition*, pages 61–70. Springer, 2006.
- [3] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1395–1402. IEEE, 2005.
- [4] Harry Blum et al. A transformation for extracting new descriptors of shape. *Models for the perception of speech and visual form*, 19(5):362–380, 1967.
- [5] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004.
- [6] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001.
- [7] Wenli Cai, Dongqing Chen, and Frank C Dachele. Vessel segmentation using vesselness and edgeness, November 24 2004. US Patent App. 10/580,742.
- [8] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.
- [9] Kunlin Cao, Kai Ding, Gary E Christensen, and Joseph M Reinhardt. Tissue volume and vesselness measure preserving nonrigid registration of lung ct images. In *Proc. SPIE*, volume 7623, page 762309, 2010.

- [10] Kunlin Cao, Kaifang Du, Kai Ding, Joseph M Reinhardt, and Gary E Christensen. Regularized nonrigid registration of lung ct images by preserving tissue volume and vesselness measure. *Grand Challenges in Medical Image Analysis*, 2010.
- [11] Laurent D Cohen and Ron Kimmel. Global minimum for active contour models: A minimal path approach. *International journal of computer vision*, 24(1):57–78, 1997.
- [12] Andrew Delong, Anton Osokin, Hossam N Isack, and Yuri Boykov. Fast approximate energy minimization with label costs. *International Journal of Computer Vision*, 96(1):1–27, 2012.
- [13] Konstantinos G Derpanis. The harris corner detector. *York University*, 2004.
- [14] Thomas Deschamps and Laurent D Cohen. Fast extraction of minimal paths in 3d images and applications to virtual endoscopy. *Medical Image Analysis*, 5(4):281–299, 2001.
- [15] Thomas Deschamps and Laurent D Cohen. Fast extraction of tubular and tree 3d surfaces with front propagation methods. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 731–734. IEEE, 2002.
- [16] Klaus Drechsler and Cristina Oyarzun Laura. Comparison of vesselness functions for multiscale analysis of the liver vasculature. In *Information Technology and Applications in Biomedicine (ITAB), 2010 10th IEEE International Conference on*, pages 1–5. IEEE, 2010.
- [17] Andinet Enquobahrie, Luis Ibanez, Elizabeth Bullitt, and Stephen Aylward. Vessel enhancing diffusion filter. *The Insight Journal*, 2007.
- [18] Nils Daniel Forkert, Dennis Säring, Karolin Wenzel, Till Illies, Jens Fiehler, and Heinz Handels. Fuzzy-based extraction of vascular structures from time-of-flight mr images. In *MIE*, pages 816–820, 2009.
- [19] Alejandro F Frangi, Wiro J Niessen, Koen L Vincken, and Max A Viergever. Multi-scale vessel enhancement filtering. In *Medical Image Computing and Computer-Assisted Intervention MICCAI'98*, pages 130–137. Springer, 1998.
- [20] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.
- [21] Hossam Isack and Yuri Boykov. Energy-based geometric multi-model fitting. *International journal of computer vision*, 97(2):123–147, 2012.

- [22] Marcel Jackowski, Xenophon Papademetris, Lawrence W Dobrucki, Albert J Sinusas, and Lawrence H Staib. Characterizing vascular connectivity from microct images. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005*, pages 701–708. Springer, 2005.
- [23] Stefanie Jegelka and Jeff Bilmes. Submodularity beyond submodular energies: coupling edges in graph cuts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1897–1904. IEEE, 2011.
- [24] JH Kinney, QC Johnson, MC Nichols, U Bonse, RA Saroyan, R Nusshardt, and R Pahl. X-ray microtomography on beamline x at ssrl. *Review of Scientific Instruments*, 60(7):2471–2474, 1989.
- [25] Pushmeet Kohli, Anton Osokin, and Stefanie Jegelka. A principled deep random field for image segmentation. 2013.
- [26] Vladimir Kolmogorov and Ramin Zabini. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):147–159, 2004.
- [27] Yiannis Kyriakou, Daniel Prell, and Willi A Kalender. Ring artifact correction for high-resolution micro ct. *Physics in medicine and biology*, 54(17):N385, 2009.
- [28] Hua Li and Anthony Yezzi. Vessels as 4-d curves: Global minimal 4-d paths to extract 3-d tubular surfaces and centerlines. *Medical Imaging, IEEE Transactions on*, 26(9):1213–1223, 2007.
- [29] Tony Lindeberg. Feature detection with automatic scale selection. *International journal of computer vision*, 30(2):79–116, 1998.
- [30] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [31] Hengameh Mirzaalian and Ghassan Hamarneh. Vessel scale-selection using mrf optimization. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3273–3279. IEEE, 2010.
- [32] Beat Münch, Pavel Trtik, Federica Marone, and Marco Stampanoni. Stripe and ring artifact removal with combined wavelet-fourier filtering. *Opt. Express*, 17(10):8567–8591, 2009.

- [33] Julia Alison Noble. *Descriptions of image surfaces*. PhD thesis, University of Oxford, 1989.
- [34] Carl Olsson and Yuri Boykov. Curvature-based regularization for surface approximation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1576–1583. IEEE, 2012.
- [35] Daniel Prell, Yiannis Kyriakou, and Willi A Kalender. Comparison of ring artifact correction methods for flat-detector ct. *Physics in medicine and biology*, 54(12):3881, 2009.
- [36] Daniel Ruijters, Bart M ter Haar Romeny, and Paul Suetens. Vesselness-based 2d–3d registration of the coronary arteries. *International journal of computer assisted radiology and surgery*, 4(4):391–397, 2009.
- [37] Robert JT Sadleir and Paul F Whelan. Fast colon centreline calculation using optimised 3d topological thinning. *Computerized Medical Imaging and Graphics*, 29(4):251–258, 2005.
- [38] Nancy M Salem and Asoke K Nandi. Unsupervised segmentation of retinal blood vessels using a single parameter vesselness measure. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pages 528–534. IEEE, 2008.
- [39] Jianbo Shi and Carlo Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.
- [40] Jan Sijbers and Andrei Postnov. Reduction of ring artefacts in high resolution micro-ct reconstructions. *Physics in Medicine and Biology*, 49(14):N247, 2004.
- [41] Olena Tankyevych, Hugues Talbot, Petr Dokladál, and Nicolas Passat. Spatially-variant morpho-hessian filter: Efficient implementation and application. In *Mathematical Morphology and Its Application to Signal and Image Processing*, pages 137–148. Springer, 2009.
- [42] Engin Türetken, Carlos Becker Przemysław Głowacki Fethallah Benmansour, Pascal Fua, and EPFL CVLab. Detecting irregular curvilinear structures in gray scale and color imagery using multi-directional oriented flux. *methods*, 16:7.
- [43] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Graph cut based image segmentation with connectivity priors. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

- [44] Ming Wan, Zhengrong Liang, Qi Ke, Lichan Hong, Ingmar Bitter, and Arie Kaufman. Automatic centerline extraction for virtual colonoscopy. *Medical Imaging, IEEE Transactions on*, 21(12):1450–1460, 2002.
- [45] M Zellerhoff, B Scholz, EP Ruehrnschopf, and T Brunner. Low contrast 3d-reconstruction from c-arm data. In *Proc. of SPIE Vol*, volume 5745, page 647, 2005.

Appendix A

Visualization of 3D Data

A.1 Maximum Intensity Projection

Maximum intensity projection is widely used in scientific research for 3D volume visualization. On the projection plane, only the voxel with the maximum along the projection ray are displayed as is illustrated in Figure A.1.

The human brain cannot perceive depth with only one frame of maximum intensity projection. Therefore, we normally use maximum intensity projection with an animation of rotation.

Figure A.2a shows the rendering result of the surface of the volume. Figure A.2b shows the same volume using maximum intensity projection. As we can see, maximum intensity projection helps us perceive the data must more efficiently.

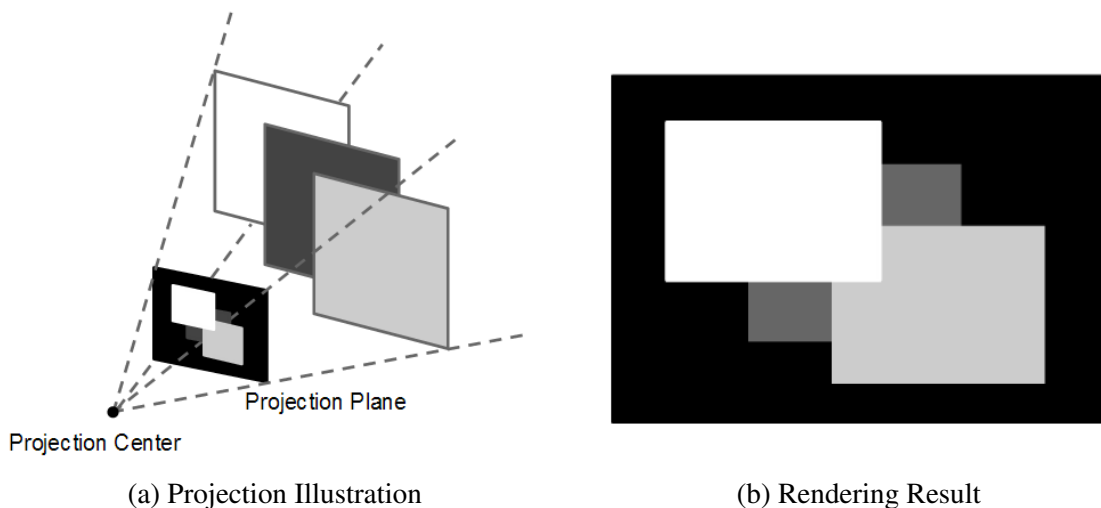


Figure A.1: Maximum Intensity Projection

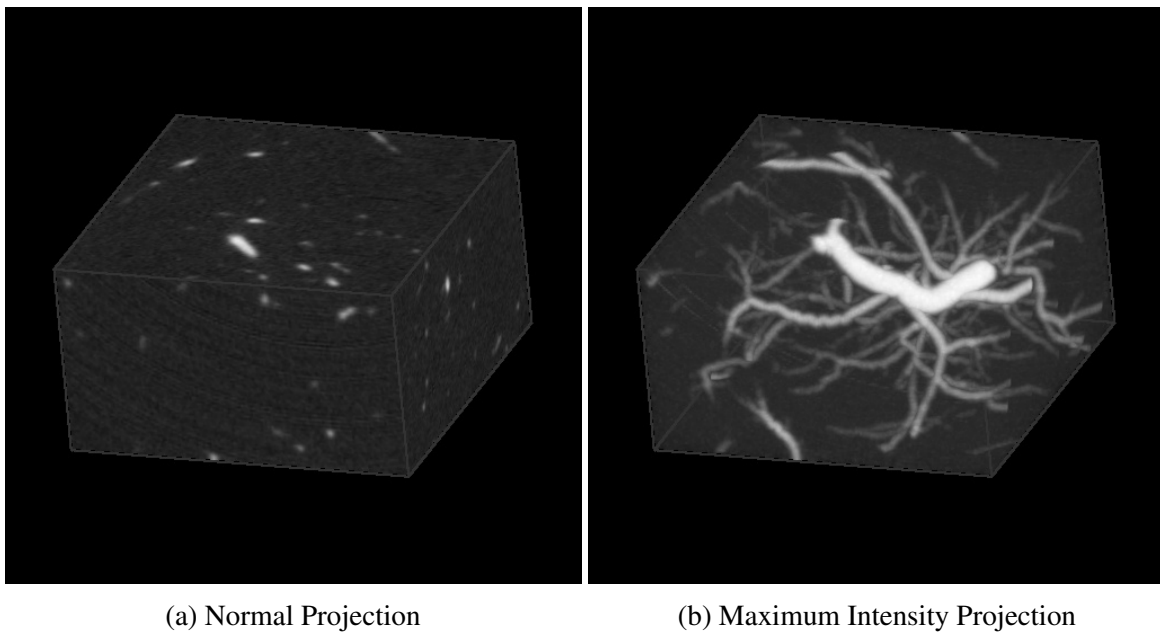


Figure A.2: Comparing Between Normal Projection and Maximum Intensity Projection

A.2 Arbitrary Cross Section

The cross section of a cube can be: 1) Triangle; 2) Rectangle; 3) Pentagon; 4) Hexagon. Some examples are shown in Figure A.3. Assume that we are getting a hexagon. We can get the intersection points of the cross plane with each side of the cube and get the hexagon $ABCDEF$ (Figure A.3c). The question is: how do we sort the intersection points so that the hexagon $ABCDEF$ can be divided into 4 triangles ABC , ACD , ADE and AEF so that they can be visualized in OpenGL. Notice that the order of the points is very important because if we need to divide them into the proper triangles.

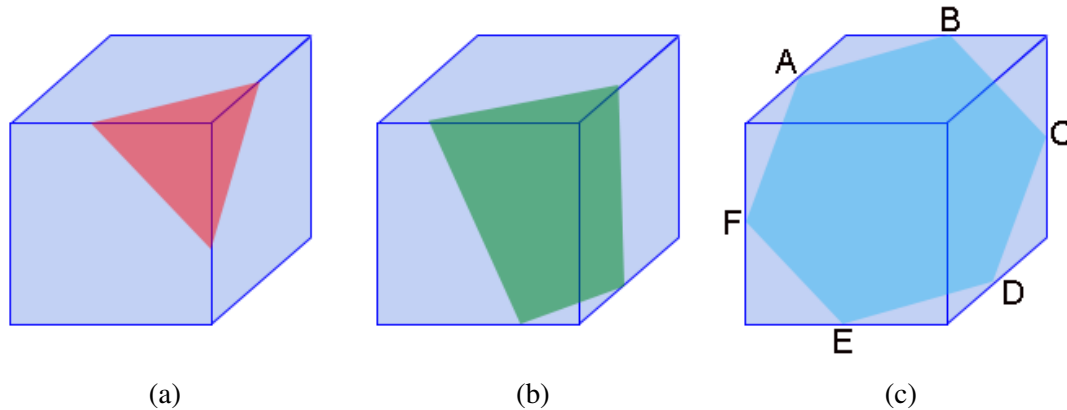


Figure A.3: Cross Section of a Box

The intersection is a convex polygon, so any sorting that works for convex polygons will work here as well¹. In particular:

- calculate the centroid (N being number of points)

$$Z = \frac{A + B + C + \dots}{N}$$

- calculate the normal of the cross section

$$n = \vec{AB} \times \vec{BC}$$

- order all points P by the signed angle \vec{ZA} to \vec{ZP} with normal n

$$\text{signed angle} == \text{acos} \left(\frac{\langle \vec{ZA}, \vec{ZP} \rangle}{|\vec{ZA}| |\vec{ZP}|} \right) \cdot \text{sign}(\langle n, |\vec{ZA}| \times |\vec{ZP}| \rangle)$$

¹The solution is provided by @HugoRune at Stackoverflow. Here is the link of the original post — <http://stackoverflow.com/questions/20387282/compute-the-cross-section-of-a-cube>

Figure A.4 show arbitrary cross sections of our data volume. This rendering technique is better than maximum intensity projection (Appendix A.1) when we want to look at the accurate voxel intensity. Maximum intensity projection is better than this if we want to get an overall perception of the result.

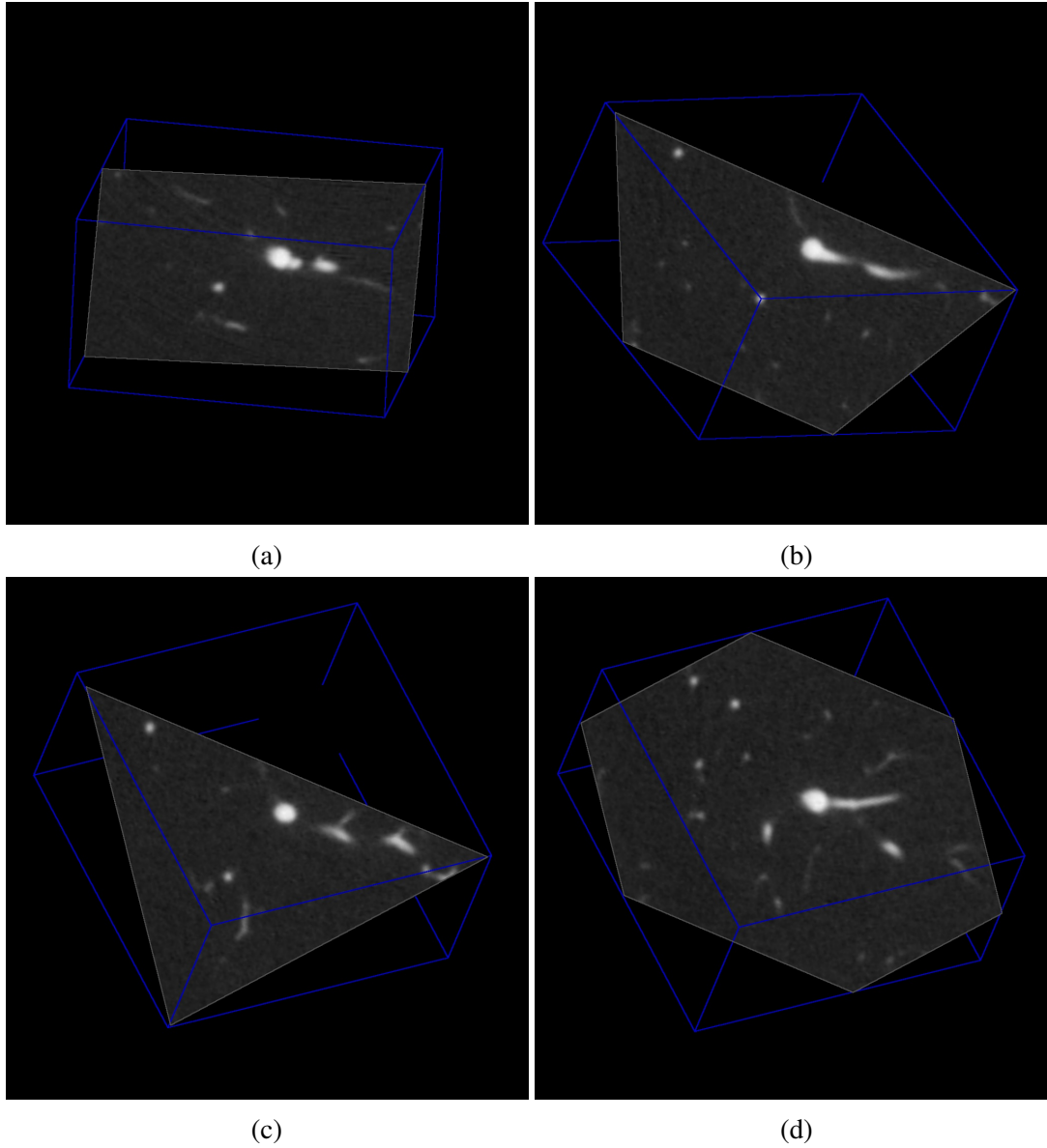


Figure A.4: Cross Section of The Vessel Volume

Appendix B

Rings Reduction

Our rings reduction algorithm is inspired by Sijbers et al. [40]. They achieve rings reduction by applying a *median and mean filter* to the image. In brief, this approach contains the following steps:

- transform the image into polar coordinates so that the rings become parallel lines in the image (see Figure B.1a)
- median and mean filtering
 - compute the average value for each row in the sliding window indicated in Figure B.1a and deduct the value by the first or leftmost value in the row
 - the result tells us how much stronger or weaker the intensity of the first column is
 - for each sliding window, we have N values from the previous step (where N equals to the number of rows)
 - compute the mean value among them and use it as the artifact templates for the first column in the sliding window
- correct line artifacts based on the set of artifact templates computed in the previous step
- transform the image back into Cartesian coordinates

We don't have to do the filtering in polar coordinate, we can achieve this in Cartesian coordinates as well. Some comparing between these two are available in [35]. We are doing rings reduction in Cartesian coordinates for two reasons. First, transformation to and from polar coordinate requires a lot of data interpolation. Interpolation will result in the loss of accuracy. Second, it is more efficient to do the computation in Cartesian coordinate because in

polar coordinate there are a lot of redundant data. Especially when it is close to the centre of the rings, several pixels are interpreted as a whole column in polar coordinates.

In order to take advantage of 3D data, we also take into consideration the neighbouring slices when doing mean filtering. Similarly, we compute the median value for each rings and get the artifact templates. Figure B.2 shows the comparison before and after rings reduction. We get reasonable result when it is far away from the centre of rings. However, the result is still not satisfying at the centre of rings.

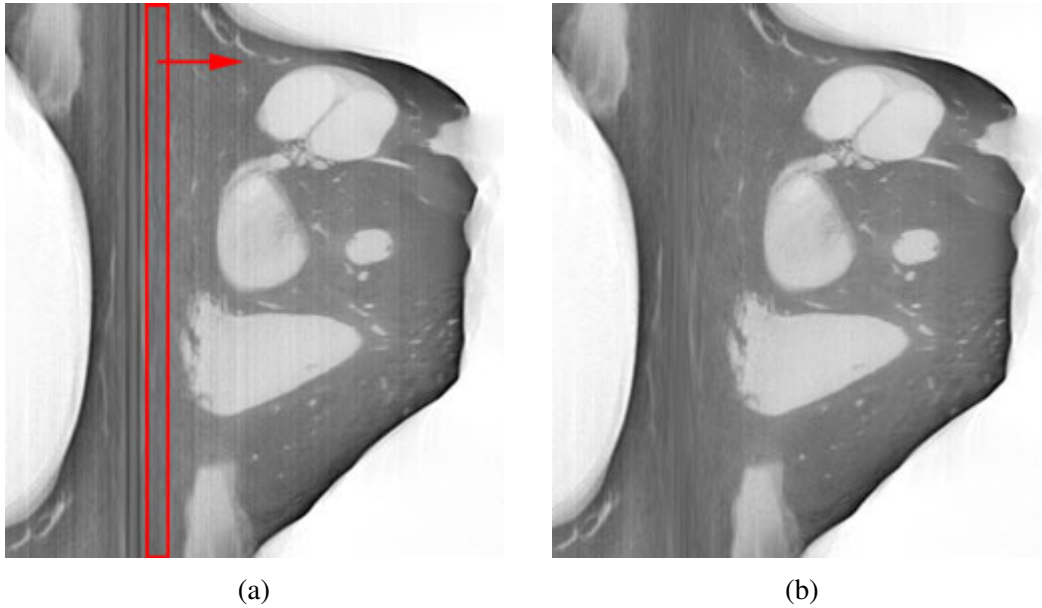
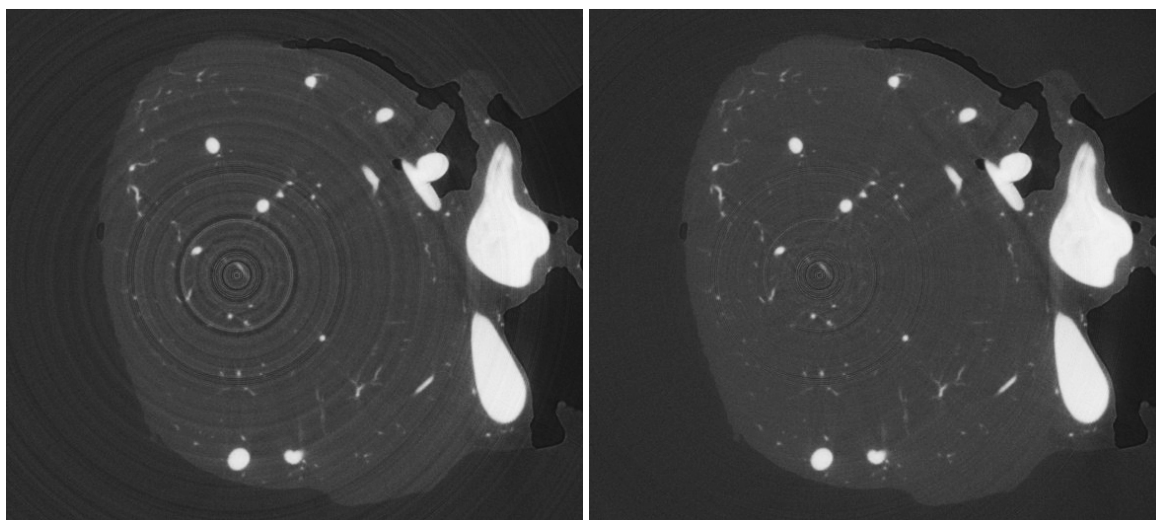
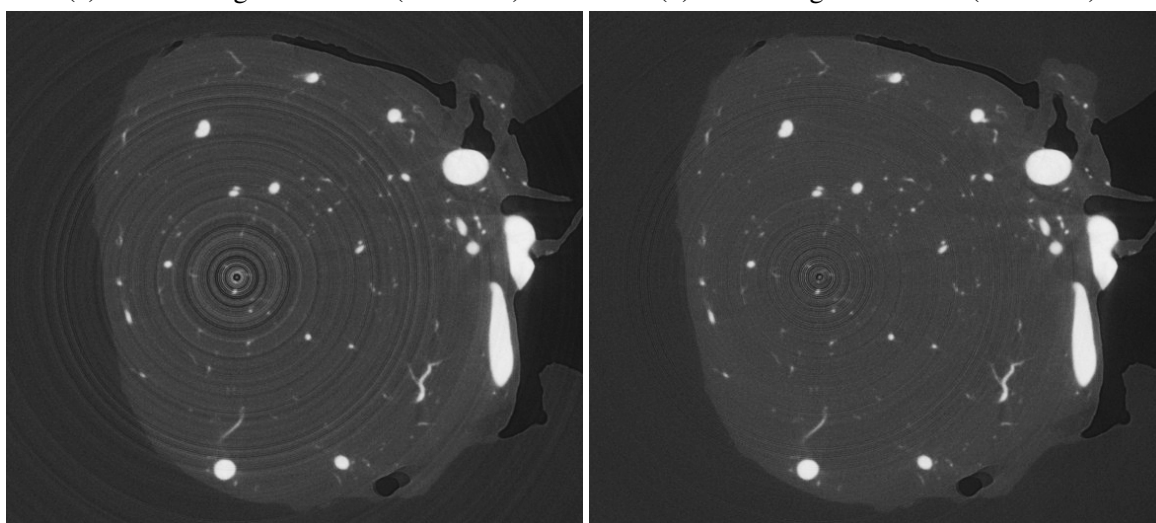


Figure B.1: Rings Reduction in Polar Coordinates [40]



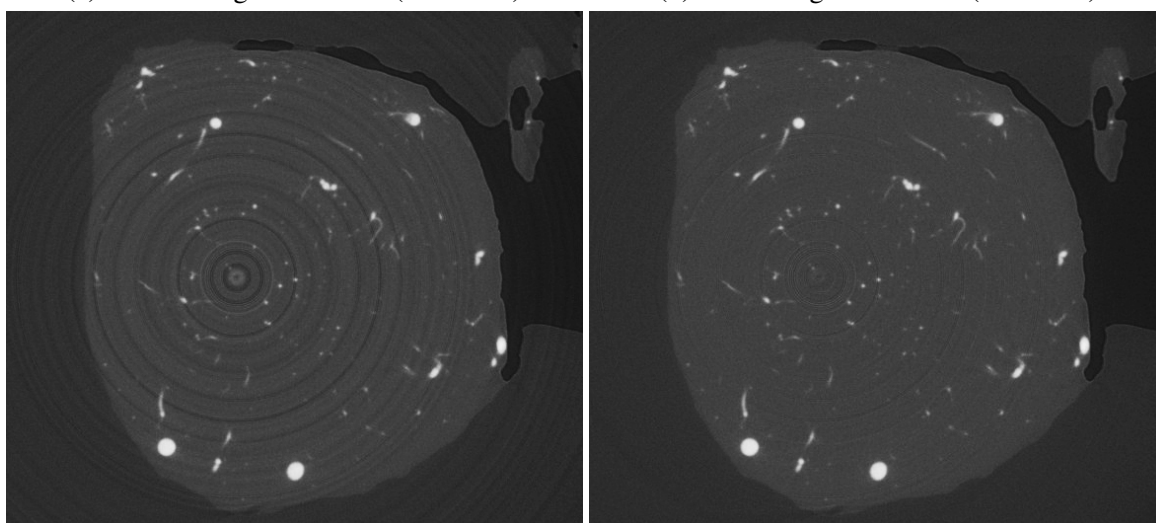
(a) Before Rings Reduction (Slice 351)

(b) After Rings Reduction (Slice 351)



(c) Before Rings Reduction (Slice 386)

(d) After Rings Reduction (Slice 386)



(e) Before Rings Reduction (Slice 494)

(f) After Rings Reduction (Slice 494)

Figure B.2: Comparison Before and After Rings Reduction

Appendix C

Eigenvalues of a Symmetric Matrix

Theorem C.0.1 *Eigenvalues of a symmetric matrix are real numbers*

Proof (MIT Open Course Lecture Nodes, [click here for the link](#))

Suppose A is symmetric and $Ax = \lambda x$. Then we can conjugate to get $A\bar{x} = \bar{\lambda}\bar{x}$. (This proves that complex eigenvalues of real valued matrices come in conjugate pairs.) Now transpose to get $x^T A^T = x^T$. Because A is symmetric we now have $x^T A = x^T \lambda$. Multiplying both sides of this equation on the right by x gives: $x^T Ax = x^T \lambda x$. On the other hand, we can multiply $Ax = \lambda x$ on the left by x^T to get: $x^T Ax = x^T \lambda x$. Comparing the two equations we see that $x^T \lambda x = x^T \lambda x$ and, unless $x^T x$ is zero, we can conclude $\lambda^T = \lambda$ is real. How do we know $x^T x \neq 0$?

$$x^T x = \begin{bmatrix} \bar{x}_1 & \bar{x}_2 & \vdots & \bar{x}_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1^2 + x_2^2 + \dots + x_n^2 = 0 \quad (\text{C.1})$$

If $x = 0$ then $x^T x = 0$. ■

Theorem C.0.2 *The eigenvectors of a symmetric matrix A corresponding to different eigenvalues are orthogonal to each other.*

Proof Let $\lambda_i \neq \lambda_j$. Pre-multiply v_j^T to $Av_i = \lambda_i v_i$,

$$v_j^T Av_i = v_j^T \lambda_i v_i \quad (\text{C.2})$$

Take the transpose of $Av_j = \lambda_j v_j$ on both side, we have $v_j^T A^T = \lambda_j v_j^T$, and we post-multiply both sides by v_i ,

$$v_j^T A^T v_i = \lambda_j v_j^T v_i \quad (\text{C.3})$$

Subtracting Equation (C.2) and Equation (C.3) yields $(\lambda_i - \lambda_j)v_i^T v_j = 0$, from which it follows that $v_i^T v_j = 0$. ■

Appendix D

Distance in 3D Space

This section is about some basics of 3D geometry. We believe they are very fundamental and easy to understand, but we failed to find good resources. Therefore, they are documented them down here for any future references.

D.1 Distance Between Point to Line in 3D

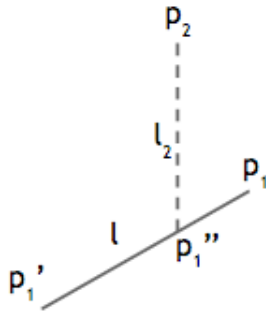


Figure D.1: Distance From Point to Line in 3D

Assume we have line in 3D, which is defined by a pair of 3D points p_1 and p_1' . We are looking for the distance from an arbitrary 3D point p_2 to this line as is illustrated in Figure D.1.

p_1'' is a point on the line and l_2 is the direction from a p_2 to p_1''

$$p_1'' = tp_1' + (1 - t)p_1$$

$$l_2 = p_1'' - p_2$$

$$l = p_1' - p_1$$

Then the goal is to determine t such that $\|l_2\|$ is minimized.

The Solution

$\|l_2\|$ reaches the minimum value if and only if l_2 is perpendicular to l .

$$l_2^T l = 0$$

which gives,

$$\begin{aligned} (tp'_1 + (1-t)p_1 - p_2)^T (p'_1 - p_1) &= 0 \\ \Rightarrow (p'_1 - p_1)^T (p'_1 - p_1)t + (p_1 - p_2)^T (p'_1 - p_1) &= 0 \end{aligned}$$

Therefore,

$$t = -\frac{(p_1 - p_2)^T (p'_1 - p_1)}{(p'_1 - p_1)^T (p'_1 - p_1)}$$

With t , we can determine the intersection point as well as the distance easily.

D.2 Distance Between Two Lines in 3D

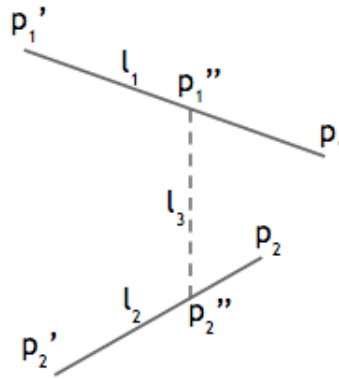


Figure D.2: Distance Between Lines in 3D

Assume we have two lines in 3D. Each line is defined by a pair of 3D points.

$$l_1 = p'_1 - p_1$$

$$l_2 = p'_2 - p_2$$

where l_1 and l_2 are the directions along the lines; and p'_1 and p_1 are two points on the first line; and p'_2 and p_2 are two points on the second line.

As is illustrated in Figure D.2, p''_1 is an arbitrary point on line 1 and p''_2 is an arbitrary point on line 2. Their relationship with p_1 , p'_1 , p_2 and p'_2 are

$$\begin{aligned} p''_1 &= t_1 p'_1 + (1 - t_1) p_1 \\ p''_2 &= t_2 p'_2 + (1 - t_2) p_2 \end{aligned}$$

Assume line 3 intersects with line 1 and line 2 on p''_1 and p''_2 respectively. Then the direction along line 3 is

$$\begin{aligned} l_3 &= p''_2 - p''_1 \\ &= [t_2 p'_2 + (1 - t_2) p_2] - [t_1 p'_1 + (1 - t_1) p_1] \\ &= (p'_2 - p_2) t_2 - (p'_1 - p_1) t_1 - (p_2 - p_1) \\ &= l_2 t_2 - l_1 t_1 - (p_2 - p_1) \end{aligned}$$

The goal is to find the smallest magnitude $\|l_3\|$.

The Solution

If $\|l_3\|$ is the smallest distance between line 1 and line 2, then line 3 should be perpendicular to both the two lines.

$$\begin{aligned} l_3^T l_1 &= l_2^T l_1 t_2 - l_1^T l_1 t_1 - (p_2 - p_1)^T l_1 = 0 \\ l_3^T l_2 &= l_2^T l_2 t_2 - l_1^T l_2 t_1 - (p_2 - p_1)^T l_2 = 0 \end{aligned}$$

That is

$$\begin{aligned} l_2^T l_1 t_2 - l_1^T l_1 t_1 &= (p_2 - p_1)^T l_1 \\ l_2^T l_2 t_2 - l_1^T l_2 t_1 &= (p_2 - p_1)^T l_2 \end{aligned}$$

Solve the linear equations in two unknowns t_1 and t_2 , we will be able to determine the intersection point as well as the shortest the distance between the two lines.

D.3 Distance Between Two Line Intervals in 3D

We can still represent an arbitrary point on a line segment with interpolation of the two endpoints. In another words, the following still holds.

$$\begin{aligned}p_1'' &= t_1 p_1' + (1 - t_1) p_1 \\ p_2'' &= t_2 p_2' + (1 - t_2) p_2\end{aligned}$$

The only difference is that since they are line intervals, both t_1 and t_2 should be within the range of $[0, 1]$.

We can still use the method in Section D.2 to calculate the distance between two lines. If either t_1 or t_2 is not in the range of $[0, 1]$, we use method of Section D.1 to calculate the distance between the each of the end points to the other line and the choose the minimum one.

Curriculum Vitae

Name: Yuchen Zhong

Post-Secondary Education and Degrees: The University of Western Ontario
London, ON Canada
2012 - 2014 M.S.

Zhejiang University
Hangzhou, ZJ China
2008 - 2012 B.E.

Honours and Awards: Graduate Research Scholarship
2012-2013

Faculty of Science Graduate Student Teaching Award
2013

Related Work Experience: Teaching Assistant
The University of Western Ontario
2012 - 2013

Research Assistant
The University of Western Ontario
2012 - 2013