Acquisition and Analysis of Mouse Brain Vasculature

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

by

Shruthi Raghavan
National Institute of Technology, Tiruchirappalli
Bachelor of Technology, 2012

June 2016
Kettering University

This thesis is approved for recommendation to the Graduate Council.

Jaerock

Jaerock Kwon
Thesis Director

Girma Tewolde
Committee Member

Ravi Warrier

Ravi Warrier
Committee Member

Abstract

Acquisition and Analysis of Mouse Brain Vasculature

by

Shruthi Raghavan

Master of Science in Computer Engineering

Kettering University, Flint, MI

Professor Jaerock Kwon, Chair

Cerebrovascular disease has been one of the major causes of deaths for many years. Microvasculature network anatomy can help biomedical researchers diagnose and relate structure and function for blood-vessel diseases that affect the brain. Angiogenesis, neural activity, and other important processes have been found to be influenced by the anatomy of vasculature networks. However, there is a huge lack of information about the structure-function relationship for vasculature in the brain. To begin unraveling the mysteries of the vasculature network in the human brain, researchers image and analyze the brain of the mouse, which has multiple similarities to the human brain in both structural development and gene expression. Due to these similarities, the mouse brain can help understand the complexities and structures in the human brain, despite being a fraction of the size and weight of the human brain. However, a complete sub-micrometer resolution dataset with morphological statistics derived for the mouse brain is yet to be made available.

In this thesis, I focus on obtaining structure of vasculature in the mouse brain at sub-micrometer resolution and on analyzing this data. The mouse brain vasculature data-sets were acquired using a technology called Knife-Edge Scanning Microscopy (KESM), which is a unique technology for automated microscopy. The Internet-Enabled Robotic Microscope (IEROM) was setup for acquisition and imaging of tissue samples using KESM. It improves the first prototype, which was called the Brain Tissue Scanner (BTS), by reducing cost

and footprint and increasing flexibility. It also connects the instrument to the Internet for remote access and information sharing among the research community. The data processing pipeline comprises the preprocessing of raw images from the IEROM to enable easier feature extraction, analysis, and visualization. I proposed a new method for preprocessing of the images and designed a data container for tubular feature representation during the course of this thesis. I also implemented an analysis engine to derive analytical information from the whole mouse brain dataset. Finally, a new approach has been proposed for processing of tubular features in teravoxel image stacks derived from light microscopy.

I used this pipeline to quantitatively analyze the entire mouse brain vasculature dataset obtained using KESM. The statistical results are stored in a web-shareable document format. I compared the statistics from two different regions of the mouse brain, namely the occipital lobe and cerebellum, which clearly indicates the usefulness of structure to deducing functionality. This type of information can also be used in research about the development of brain structure, behavioral patterns and response to stimuli. The models for mouse brains and correlate the results with the human brain. Such models can be useful in diagnosis and analysis of disease development and may lead to methods of their prevention and/or cure. The pipeline I have designed can be used to extract data to use and publish in structural micro-vasculature atlases for mouse brains at different stages of development for studying and diagnosing related diseases. This information can be used by biomedical researchers to derive medical conclusions.

Acquisition and Analysis of Mouse Brain Vasculature

I dedicate this thesis
to my unconditionally supportive and loving husband
Mr.Rahul Rangarajan,
to my parents & my brother
Mr.HKN.Raghavan, Mrs.Sowmya Raghavan, & Mr.Sai Srinivas
and to my father-in-law & sister-in-law
Mr.R.Rangarjan & Ms.Rashmi Rangarajan.
Thank you all, for always believing in me and being there for me.

Contents

List of Figures

List of Tables

Acknowledgments

The research included in this thesis could not have been performed if not for the assistance, patience, and support of many individuals. I would first like to thank my thesis advisor Professor Jaerock Kwon of the Electrical and Computer Engineering Department at Kettering University for the opportunity to work in his project. Prof.Kwon was always there for me as a guide and mentor and gave me independence to do my research while steering me in the right direction whenever he felt I needed it. Prof.Kwon's office door was always open for any clarifications and questions I had. His guidance has helped me to become a better writer and researcher and I sincerely thank him for his confidence in me. I would also like to acknowledge Professor Ravi Warrier of the Electrical and Computer Engineering Department at Kettering University as a second reader to my thesis. I would like to thank him for his passionate participation and valuable comments on this thesis.

This research would not have been possible without the assistance of 3Scan Inc., who constructed the new instrument, and that of the Brain Networks Laboratory at Texas A&M University, who built the foundations for the imaging technique that made this instrument possible. I want to also thank Matt Goodman and Gerald Spencer at 3Scan Inc. They shared a lot of valuable information about the setup of the image acquisition instrument. I would like to acknowledge the conversion of the stitched dataset into sub-volumes by Ms.Akanksha Ashwini. I want to thank Dr.Junha Lee for his advice, mentoring, and for all the valuable guidelines he shared from his experience.

I am very thankful to all my friends at 620, and Gerald Kozlowski, Taajwar Bey, Hoshing Kwong and Nicholas Hayes, for making my journey enjoyable and fun. They were constantly there for me and gave me so much positivity while I worked towards my thesis. Finally, I want to express my very profound gratitude to my family, especially to my husband Rahul Rangarajan, for providing me with unwavering support and continuous encouragement throughout my study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Part I

Introduction

Chapter 1

Introduction

This part provides a general overview of the contents of the thesis. The intent is to provide the reader a glance at the research problem, work done towards a solution and the supporting publications.

Section 1.1 explains the motivations behind this work and research. Section 1.3 describes the goals of this thesis. Section 1.4 enumerates the contents of this thesis. Section 1.5 lists the publications that were secured during the work of this thesis and those pending approvals. Finally, Section 1.6 describes the organization and structure of the remaining thesis for the interested reader.

1.1 Motivations

Structural connectomes are researched today with an intent to map out the interior circuits of the brain. This can help biomedical researchers understand the structure-function relationship and also diagnose disease by understanding the various pathways affected by a cause. While many of these studies focus on the neuronal networks, the vasculature and microvasculature in the brain are also gaining momentum in research today.

Cerebrovascular disease has notoriously been one of the major causes of deaths for many years [34, 31, 30]. So it is imperative to study this system and map out the structure to disease probabilities to help in detection and diagnosis. Moreover, detailed analysis of vasculature structure in healthy specimen can be used as golden reference model for finding the cause of anomalies in other specimen.

The micro-vascular network anatomy can help biomedical researchers in figuring out the circulatory system function.Physiological systems in micro-vessels are closely related to their micro-vascular architecture. Blood flow regulation, angiogenesis, structural adaptation of vascular beds and blood flow response to neural activity are all influence by the anatomy of micro-circulation. The anatomy even influences the interpretation of hemodynamic based functional images[39]. It has been pointed out that future progress will require not only investigation of molecular and cellular mechanisms but also network-level studies that show how these mechanisms are integrated in a physiological context[74]."

A vast variety of methods and approaches have been developed for vascular extraction, analysis and modelling with increasing complexity[40]. It is important to have a structured pipeline that can incorporate these approaches and create a work environment for taking a tissue sample and processing it all the way until it is shared with the medical community in a useful way.

Moreover, we have identified the need for a standard data container and analysis algorithm targeted at vasculature geometries for morphometric analysis. This is because the vasculature structure is different from neuron structure in significant ways and a analysis algorithm targeted at both these structures has not been developed yet.

1.2 The Rodent Brain in Research

The brain is very similar the central processing unit of a mammal. It calculates, remembers and controls structure and function of the organism. Studying the rodent brain to deduce and understand the more complex human brain is a practice that has been around for decades and there is a good reason why. It is natural to question if a tiny rodent brain that weighs only a small fraction of the human brain can help understand the complexities and structures in the latter. The answer is yes, it can. The rodent brain, it turns out, is very much like the human brain in terms of structural development and gene expression. This information can be used in research of important subjects the development of brain structure, behavioral patterns and response to stimuli. We can use the models for rodent brains and correlate the results with the human brain. Such models can be useful in diagnosis and analysis of disease development and may lead to methods of their prevention and/or cure.

All mammalian brains start off as a series of three swellings in the embryonic neural tube called the brain vesicles that eventually form the fore-brain, mid-brain, and hind-brain vesicles. For study of the details of brain connections and function, rodent can be used because they share even fine details in the structure. The rodent brain and human brain have all the basic features of a typical mammalian brain. Although the human brain has special features like a massively developed cerebral cortex, many important studies on the cause of human brain disease have been based on work on the rodent brain. For example, development of new drugs for treatment of Parkinson's disease are based on experiments on the rat brain[71].

1.3 Objectives

General Objectives

- Setup the Internet Enabled Robotic Microscope(IEROM) for imaging tissue specimen seamlessly and document the process for reproducible research.

- Develop a fully automated image processing pipeline for shorter time to availability of the data-sets imaged by the IEROM to the biomedical research community.

- Contribute to study of microvasculature by development tools to process IEROM data-sets and to analyze and store morphometric properties for reference.

Operational Objectives

- Study the Knife Edge Scanning Microscopy(KESM) and the initial prototype based on this technology, namely the Brain Tissue Scanner(BTS) to better understand IEROM.
- Setup of the IEROM:
  - Assembly of the main components of the IEROM including wiring and electrical connections, compressed air paths, water circulation paths and connections to the controller server.
  - Setup of samples for imaging and calibration of the instrument for a given imaging sample.
  - Detailed documentation of the process to be followed and the necessary precautions.
- Study various vascular detection and tracing algorithms for three dimensional data-sets.
- Come up with new algorithms for the pipeline that will improve upon the existing algorithms and work efficiently with data-sets imaged using the KESM principle.
- Create a Vascular Data Container and Framework that has the following features:
  - Extensible for addition of structural components
  - Easily used in programs that read, write or operate on vasculature as well as neuronal data

1.4 Thesis Contributions

In this thesis, the IEROM instrument was setup and procedures were documented in detail. The controller software of the IEROM was debugged and fixed locally at multiple points in the setup to help resolve local issues quickly. The technology behind the microscope, namely KESM, was extensively researched and the initial prototype was studied to understand the mechanism of imaging and the nature of data that will be generated by the IEROM.

Novel algorithms were developed for pre-processing and post-processing of vasculature data. The process of going from image samples to final data-set has been automated for the IEROM to enable fast processing of the data generated by the microscope. The input of the designed data processing pipeline is the digitized tissue samples available as individual images of slices organized according to their column. Then, selective normalization technique[38] was implemented to normalize the images and remove noise. Once clean data was obtained, batch processing was setup to process the images by feature extraction. A thorough study of existing techniques led to the choice of a particular tracing algorithm that was efficient in time while reasonable in accuracy. After the feature extraction is completed from the volume data, the resultant traces are analyzed to form a coherent set of comparable quantitative data. So this way, the data processing pipeline has been setup from the tissue slices to the quantitative statistics and features that are derived from it. Many changes have been proposed that can be incorporated to make this pipeline more flexible and enhance its capabilities.

Figure 1.1 shows the tasks in the data processing pipeline and the order of tasks. This shows the processing steps that will enable sharing of the data imaged by the IEROM with biomedical researchers around the world. Some of the modules that have been discussed and planned but are part of future work are also listed in this figure and indicated as such. The stitching algorithm mentioned in the pipeline and mentioned in Appendix A is color coded separately to indicate successful implementation with pending validation.

List of Technical Contributions

- A tissue extraction algorithm was developed to specifically remove the additional regions that are present on either side of the tissue in images obtained using the KESM method.

- Selective normalization was implemented to be part of the pipeline and the parameters adjusted according to the whole mouse brain vasculature data-set used throughout this thesis.

- Batch processing was done to obtain the traced features (vasculature, in this case) for further analysis. The teravoxel volume was subdivided into manageable unit volumes and traced to achieve this.

- A data container, called *Tube Morphology Container* data container was designed to hold any tubular morphology. It has been tested with vasculature and neuronal data both from the IEROM and from other scientific sources. This is extensible and can be used for morphometric analysis, format conversion and also as data representation during tracing of tubular structures.

- An analysis engine was built based on the Tube Morphology Container for calculation and storage of morphometric data for traced volumes of tubular structures. The resultant statistics are stored in a web-distribution compatible format.

- The data pipeline and the post-processing tools were tested on the mouse brain vasculature data and the results have been presented in this work.

- A method to combine sub volumes of traced data into larger volumes has been proposed. It has the potential to help mitigate the issue of teravoxel visualization. This work has been presented as proof of concept and further validation has been noted as an important step to realize this in the future.

1.5 List of Publications

- Shruthi Raghavan et al. “Internet enabled robotic microscope powered by knife-edgescanning microscopy“. In:Neuroscience Meeting Planner, New Orleans, LA: Societyfor Neuroscience.2015.

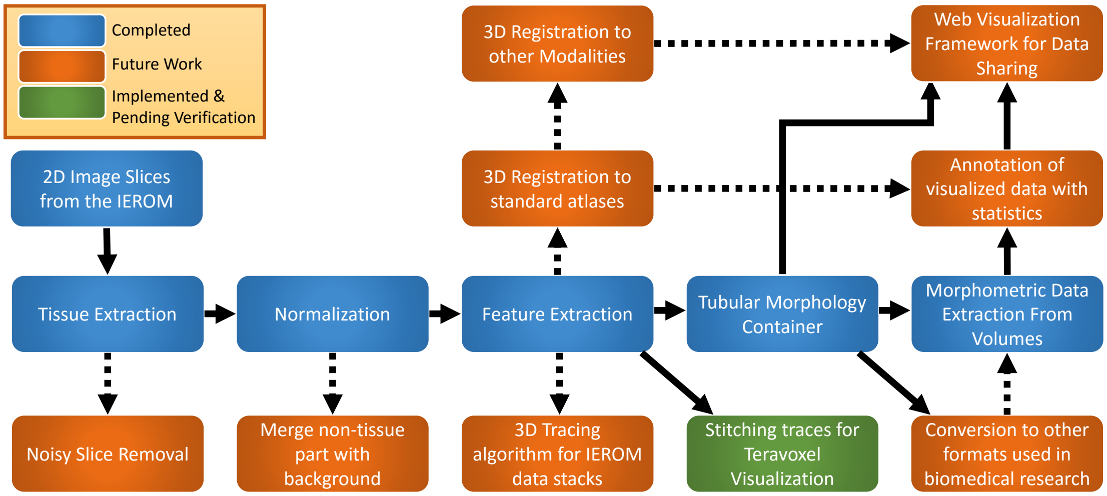

Completed
Future Work
Implemented & Pending Verification
2D Image Slices from the IEROM
Tissue Extraction
Normalization
Feature Extraction
Tubular Morphology Container
Morphometric Data Extraction From Volumes
3D Registration to other Modalities
3D Registration to standard atlases
Web Visualization Framework for Data Sharing
Annotation of visualized data with statistics
Noisy Slice Removal
Merge non-tissue part with background
3D Tracing algorithm for IEROM data stacks
Stitching traces for Teravoxel Visualization
Conversion to other formats used in biomedical research

Figure 1.1: An image pipe showing different possible steps in the image processing pipeline and the ones highlighted in blue are the implemented steps.

- Shruthi Raghavan and Jaerock Kwon. “Fully Automated Image Preprocessing for Feature Extraction from Knife-edge Scanning Microscopy Image Stacks - Towards a Fully Automated Image Processing Pipeline for Light Microscopic Images“. In:Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies. 2016, pp. 93100.isbn: 978-989-758-170-0.doi:10.5220/0005708700930100

- Shruthi Raghavan and Jaerock Kwon. “Divide, Conquer, and Combine Algorithm to-Trace Complex Tubular Structures from Teravoxel-Sized Microscope Image Stacks“.In:2016 IEEE International Symposium on Biomedical Imaging: From Nano to Macro. IEEE. 2016, doi:10.13140/RG.2.1.3539.9440

- Shruthi Raghavan, and Matt Goodman and Todd Huffman and Cody Daniel and Corey Montieth and Jaerock Kwon. “Internet-Connected High-Throughput and High-Resolution Three-Dimensional Tissue Scanner to Enable Large-Scale Automated Histology“ ***Submitted.***. In: 2016 IEEE International Conference on Imaging Systems and Techniques. IEEE. 2016.

1.6 Thesis Structure

This thesis has been divided into two logical parts. The first part introduces IEROM and the second part discusses the data processing pipeline for the digitized tissues. The first part describes the technology called Knife Edge Scanning Microscopy (KESM) which is used in the IEROM. There is also a detailed discussion on the various components of IEROM, the installation and a short guide to operate the instrument. The latter part describes in detail each implemented step of the data processing pipeline, that is shown in Figure 1.1. It is important to note that during the course of this thesis, two different data-sets were used, both of mouse brain vasculature. This was done in the interest of time to enable parallel work on multiple computers. Also, to setup the system as a whole including the instrument and the imaging pipeline, both the data-sets used in this thesis are those imaged by the Brain Tissue Scanner in the Brain Networks Laboratory at Texas A&M University in 2008.

Part II

Image Acquisition : Principles and Methods

Chapter 1

Background

Image acquisition is the process of capturing an image of the object of interest and converting it into a manageable entity [51]. In this chapter, an overview of the different image acquisition techniques for medical imaging and their applications are reviewed. Image acquisition for medical research using a light microscope, specifically a bright-field microscope [52] is discussed and the patented automated imaging method KESM, is described.

1.1 Medical Image Acquisition - An Introduction

There are multiple ways to image the internal structure of various organs today. Some of the imaging techniques have been listed and described in this section. This discussion is based on scale of imaging to illustrate the importance of the resolution and amount of information that can be obtained by these different techniques.

Macroscopic scale refers to the range of resolutions that are visible to the naked eye, starting from around millimeter resolution and covers up to hundreds of centimeter resolution . Some imaging techniques that fall under this category are Computed Tomography (CT) [35], Magnetic Resonance Imaging (MRI) [20], Positron Emission Tomography (PET) or PET-CT [5] and Ultrasound [17]. These are extremely useful in medical diagnosis to detect abnormalities that are at this scale. However, detailed medical analysis and structure and functional evaluation of biological networks requires higher resolution data sets.

Microscopic scale refers to the range of resolutions that aren't visible to the naked eye. This includes nano-meter and higher resolutions. The most important technique in this category is electron microscopy. Dense neural circuit reconstructions of small volumes of brain tissue by electron microscopy (EM) gives highly detailed structural information at even nanometer-scale resolutions but this process is very slow. Light microscopy can bridge the gap between nanometer resolution and the macroscopic data. Sub-micrometer images of medical specimen can be acquired by light microscopy (LM) methods. There are many flavors of LM methods such as bright-field LM and dark field LM, confocal LM, and fluorescence microscopy. The traditional LM image acquisition techniques involved creating tissue slides and iteratively examining each one before capturing its image. Such labor-intensive procedures greatly limit the throughput of data collection as the size of the specimen increases. This called for automation of LM instrumentation and the introduction of automation has standardized and dramatically increased the throughput of anatomical studies [54] using LM.

1.2 Knife Edge Scanning Microscopy (KESM)

KESM is one of the few automated physical sectioning based imaging techniques that allows us to view and image thick tissue specimen in a relatively small amount of time (approximately 7mm^{2}s^1). Three-dimensional light microscopy in medical imaging needs continuous automated sectioning method to automate the process. This can be achieved by either optical sectioning or physical sectioning using a suitable sample and movement stage. However, to obtain high-resolution volumetric tissue structure data at high throughput it is preferred to use physical sectioning. Optical sectioning is disadvantageous because of the depth resolution limitation and the trade-off between signal quality and disruptive background noise such as tissue data from out-of-focus imaging planes.

KESM Principle

Knife Edge Scanning Microscopy is the technology of slicing and imaging the section simultaneously being sliced with a high resolution and high sensitivity digital line camera. Since the slicing and images happens simultaneously the overall throughput of the system is high. Each slice is an aggregation of multiple line images.

The tissue sample being sliced by the knife is imaged just above the knife edge by a powerful line scan camera. The image capture mechanism is triggered based on the encoded position of the tissue being sliced, which ensures that every sectioning results in an image capture. The controller synchronizes the slicing and imaging to achieve registration of the images captured.

To slice the tissue at sub-micrometer thickness, the embedding and preparation process of the tissue is done in a specialized way described in Section 3.1. A diamond knife is used to slice the tissue with high precision and less wrinkling. The diamond knife also forms a part of the illumination path from the LED source to the objective as explained below.

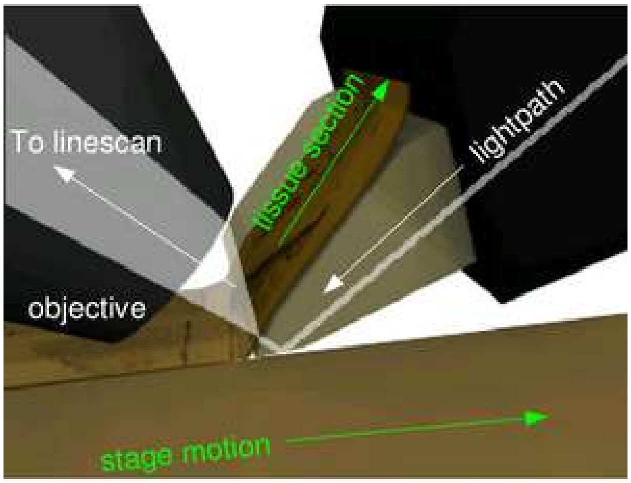


Figure 1.1: Imaging principles of the KESM. The principal of operation of KESM is illustrated. The objective and the knife is held in place, while the specimen affixed on the positioning stage moves (arrow with solid line) and gets scraped against the diamond knife, generating a thin section flowing over the knife (arrow with solid line). Line-scan imaging is done near the very tip of the knife. (Adopted from [14])

In physical sectioning microscopy, the imaging can be done by transmission illumination or reflection illumination. However reflection illumination cannot image non-reflective stains such as India Ink and also has lower throughput. This is the reason that the transmission illumination method is preferred mode of illumination for KESM based automated tissue scanning. The light source in the first prototype was a laser illumination source.

For imaging using the KESM, the laser light beam is made incident on the back surface of the diamond knife, which acts as a collimator for the light. The collimated light beam then passes through the tissue with minimal scattering and into the objective, from where it travels through the optic train before it is captured by the line scan camera. The light path is shown in Figure 1.1.

The Knife-edge scanning preserves image registration throughout the depth of the tissue block and physically sections the tissue. It does not suffer from issues such as back-scattering of light and bleaching of fluorescent-stained tissue below the knife. The KESM can image a $1cm^3$ tissue block in approximately 50 hours at a resolution of $0.6\mu m \times \ 0.7\mu m \times 1.0\mu m$. The high-throughput and high-resolution tissue scanner based on KESM has produced many data sets at sub-micrometer resolution till date.

1.3 Brain Tissue Scanner

The Brain Tissue Scanner (BTS) [46] was developed and patented by Bruce.H.McCormick in 1999. It is the first prototype of automated microscope, which was designed to digitize thick tissue samples by using KESM based at high throughput and high resolution.The instrument setup and its components are shown in Figure 1.2.

In this thesis, the BTS is explained by dividing into two logical subsystems, namely the Imaging subsystem and positioning and ultramicrotome subsystem. The specifications are listed as described in the original work [47] for completeness and description is added when required.

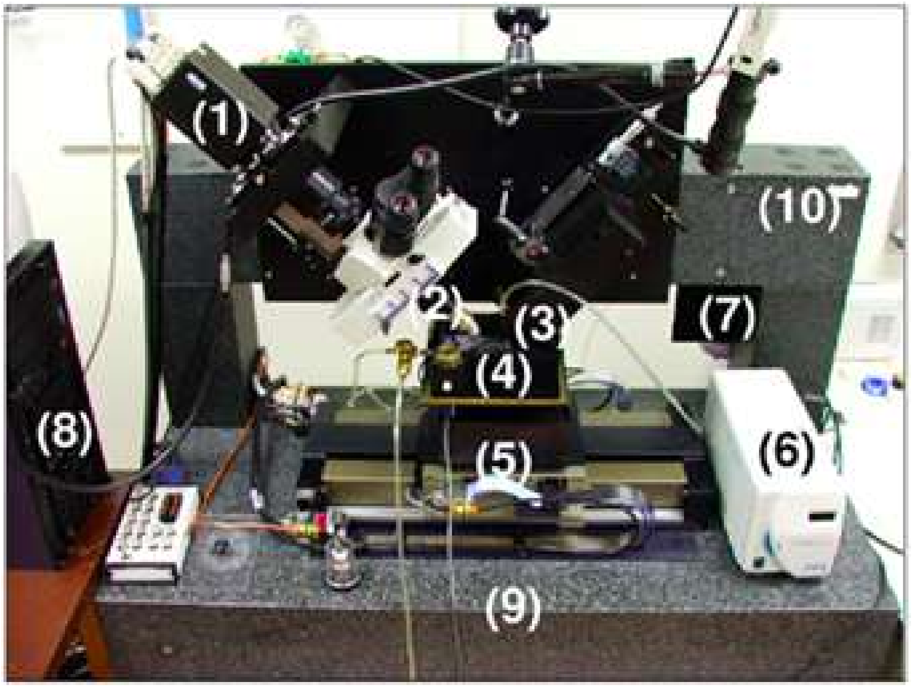


Figure 1.2: The Knife-Edge Scanning Microscope and its main components are shown: (1) high-speed line-scan camera, (2) microscope objective, (3) diamond knife assembly and light collimator, (4) specimen tank (for water immersion imaging), (5) three-axis precision air-bearing stage, (6) white-light microscope illuminator, (7) water pump (in the back) for the removal of sectioned tissue, (8) PC server for stage control and image acquisition, (9) granite base, and (10) granite bridge. (Adopted from [16])

Imaging Subsystem

The imaging subsystem consists mainly of the line scan camera, objective, illumination source and camera interface cards for image capture. Either a Dalsa CT-F3 monochrome camera at a maximum line rate of 44kHz or a Dalsa CL-T5 color camera with a line rate of 10.7kHz were used in BTS. Figure 1.2 shows the water bath in which the tissue sample is placed, indicating that the imaging is done under water. The reason for this is two-fold; to make sure the tissue samples being cut are removed from the vicinity of the specimen block being imaged and for having a larger field of view. More details on the advantages of underwater imaging are explained in Chapter 2.

The field of view (FoV) of an objective is the diameter of the largest circular area in object space that can be simultaneously viewed with specified spatial integrity. This can be expressed as the ratio of the objective diameter to the magnification. The linear array of a line-scan camera sensor images a line in the object plane of this diameter. The Numerical Aperture(N.A) of an objective is calculated using the following formula.

$$N.A = n \sin\theta \tag{1.1}$$

, where n is the index of refraction of the object medium, and 2θ is the angular aperture on the object side. Clearly, the numerical aperture increases as the refraction index of imaging medium increases (like water). It is known that the minimum distance between two resolvable points in the object plane is inversely proportional to the numerical aperture. Thus points in the plane that are closer are resolvable in a water immersion imaging setup. It is possible to argue that a higher refractive index medium can be used to enhance this effect, like oil immersion. However, water immersion objectives can achieve this result without much scattering and loss, leading to higher contrast in the digitized specimen images [8], which is of utmost importance.

For the illumination source, a laser line generator is used. The exit port of the laser line generator is behind the diamond knife, and that is where the optical chain starts in the BTS. The light then passes from the exit port through the diamond knife collimator and enters the microscope objective and through the optic train to the lens of the line scan camera. The image capture system uses three camera interface cards, two for the monochrome camera and one for the color camera. These boards are mounted on a server and receive data from the line scan camera.

Positioning and Ultramicrotome Subsystem

This subsystem consists of the stage, the ultra-microtome (diamond knife) and the specimen carrier. The initial BTS prototype's precision positioning system consisted of two stages, one for XY motion and the other for lift. Together, they could achieve motion in all three directions that is required for automatic slicing of the tissue block. The ultra-microtome in the BTS was a diamond knife that was custom cut and made and one sharpening was sufficient for sectioning of a whole mouse brain specimen at 0.5μm and 250nm XY resolution.

The specimen carrier in the BTS was a unit that was affixed on the lift stage. It consisted of a specimen loading receptacle for the tissue block, a rectangular water-containment trough for water-immersion imaging with a water inlet and outlet for continuously filtering tissue samples from the imaging medium. The design of the specimen carrier unit was opaque and difficult to see the imaging process for monitoring the sectioning that is happening in it. This makes it difficult to detect any errors during operation. Moreover, the design help more volume of water due to the rectangular shape and reached close to the load limit of the z-axis stage.

Chapter 2

The Internet Enabled Robotic Microscope (IEROM)

The BTS, first generation prototype based on KESM has already enabled important medical research via whole brain explorations of neurons, vasculature and cell structures of a mouse brain. The instrument also successfully imaged the neurons in the octopus brain at sub micrometer resolution [16, 13]. This chapter describes the second generation prototype based on the KESM.

2.1 Overview of the IEROM

The Internet Enabled Robotic Microscope (IEROM), the second generation of microscope based on KESM, is a low-cost and more robust version of the first prototype. It aims to overcome the limitations of the first generation prototype by making the instrument less expensive, flexible, less bulky and occupy smaller footprint. The cost is reduced by changes such as using LED illumination instead of laser illumination. The optics train design is modular and flexible. This provides an easily operable, flexible platform for biomedical researchers across different domains such as neuroscience and vascular research. The IEROM promises to be commercially viable and indispensable to biomedical researchers. A logical pipeline to obtain microscopy data and share it using the IEROM is shown in Fig. 2.1.

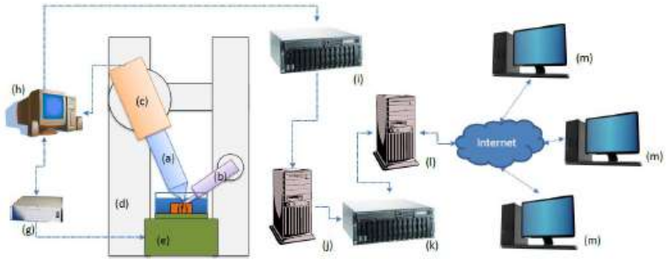


Figure 2.1: System Overview. (a)-(d) The tissue scanner: (a) microscope objective, (b) diamond knife assembly, (c) line scan camera, (d) rigid frame. (e),(f) The movement component: (e) precision positioning system, (f) tissue block in the water container. (g),(h) The controller/imaging system: (g) controller, (h) stage control/image capture station. (i)-(l) The image processing: (i) disk array for raw captured data, (j) image processing station, (k) disk array for processed data, (l) visualization/database station. (m) Computers over the network.

2.2 IEROM Hardware

The hardware setup for IEROM comprises of three main systems that are described in detail in this section. Some of the main components of this system are shown in Fig. 2.3.

3-Axis Precision Motion Stage Controller

The IEROM uses the Aerotech ANT130-060-L-Z system with 300nm accuracy and 2nm resolution for lift. For the XY motion, ANT-130-060-XY stage with 1nm resolution and 250nm accuracy is used on top of which the lift stage is placed. Aerotech linear motion stage controllers are used to drive this system. The main axis along the direction of slicing is controlled by the Aerotech HPe controller, which is responsible for triggering the image capture based on its encoded position. The other two axes are each controlled by daisy-chained Aerotech CP controllers. The interconnection between these components is as shown

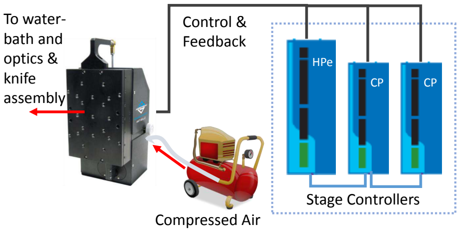


Figure 2.2: Stage and Controller Logical Connections. The air compressor unit is for the air bearing in the precision positioning stage, and the daisy-chained controllers operate the stage and trigger the image capture based on stage position.

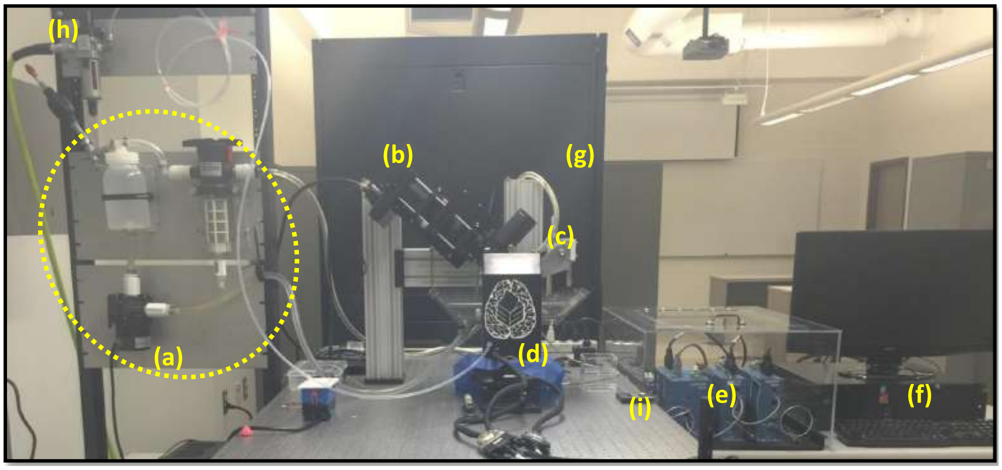


Figure 2.3: The main system components of the IEROM.*(a)* The water filtration and flow regulation system. *(b)* The optics train for imaging in the IEROM fixed to heavy support. *(c)* The knife holder module fixed to the same support as *(b)*. *(d)* The three-axis precision positioning stage from Aerotech supporting the water bath on the lift axis. *(e)* The stage controllers for each of the axes are daisy-chained. *(f)* The PC used to run the IEROM Control Client for instrument operation. *(g)* The server rack housing the Network Attached Storage (NAS), cluster servers, and a main image processing server. *(h)* The regulator that purifies and regulates the air form compressor (not shown) before feeding into the lift stage. *(i)* An example tissue sample affixed to the kinematic sample holder, ready to be placed in the water bath for slicing.

in Fig. 2.2. The compressed air required to operate the stage is provided using a compressor and a purifier-regulator. The purifier-regulator (not shown in figure) is used to ensure that the quality of the compressed air meets the specifications of the stage. We use the SMC AWD30-N02DE-Z micro mist separator regulator with $\frac{1}{4}$ in NPT (National Pipe Thread

Taper) thread. A custom interface board is used between the HPe controller and the PC frame grabber, which is connected to the camera by a Camera Link interface [4]. The camera trigger signal is generated by the controller when the stage is moving in the slicing direction. This ensures the capture and storage of every slice of tissue.

Optic Train and Knife Assembly

The optic train in KESM consists of a line scan camera, a water immersion objective, and the diamond knife and knife assembly. The light source in the IEROM is an LED illumination source. Dalsa HA-80-04K40, a monochrome TDI line scan camera with a Camera Link interface, is used in conjunction with a high resolution water immersion objective (Zeiss W N-Achroplan 10x/0.3 M27 or W N-Achroplan 40x/0.75 M27). The optics subsystem has a wide variety of options available and there is a design provision allowing the flexibility to choose which objective or configuration is used. The optics train is designed to allow room for fluorescence use in the future apart from being compatible with many objectives. Fig. 2.4(a) shows the entire setup of the optic train resting on a stable platform supported by dense, sand-filled pillars.

For focus, the camera and objective must be aligned with the surface of the diamond knife. The top surface of the knife is usually at 45° to the horizontal plane. Alignment in 6 axes is achieved using a precision angle bracket and an angular positioner. Initial setup involves slicing sequentially and adjusting the positioner to align the system and establish a focal plane. After the knife assembly is aligned with the optics and its position is registered by the control software, the tissue is continuously imaged just above the knife edge. The imaging process is illustrated in Fig. 2.4(b). The light from the LED passes through the diamond knife, which acts as a collimator and enables imaging of the tissue as it is being sectioned.

The tissue is placed on a kinematic sample holder, which is shown in Fig. 2.4(c). It has fiducials to register the position of tissue and ensures registration even in case of removal during slicing or any other interruptions by magnetic coupling to the bottom plate. A diamond knife, 2mm (for higher magnification) or 5mm (for high speed and high resolution imaging) in width is used to slice the tissue as the stage moves. Imaging happens by means of transmission illumination from the light source that is under the knife. The diamond knife acts as the collimator to this light.

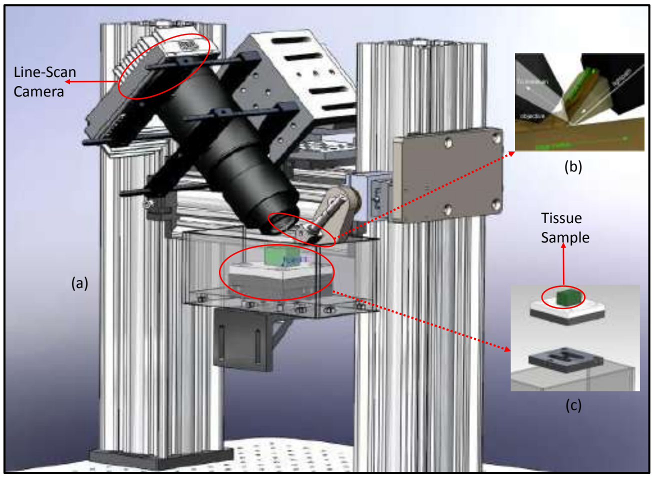


Figure 2.4: Optic Train and Knife Assembly. (a) shows the Optic Train in position along with the tissue holder and the knife assembly. The line scan camera at one end of the optic train is labelled in the figure. It consists of provisions to change objective and enough room for fluorescence. (b) The process of imaging in KESM and the flow of light for imaging near the right edge. (c) Kinematic tissue sample holder with magnetic coupling for better registration.

.

Water has many advantages as an imaging medium. It offers superior light gathering compared to air, enabling higher resolution imaging using special objectives. The water bath, however, cannot be left stagnant due to various reasons, including the compromise on purity and the eventual clouding and hindrance to imaging from the cut slices of tissue. Thus both the initial prototype and the IEROM are equipped with a water pump and filtration subsystem to maintain water flow and purity. One disadvantage of water as a medium is the corrosion on ferrous components. This can be mitigated with use of proper material for longevity for underwater use.

Water Pump and Filtration System

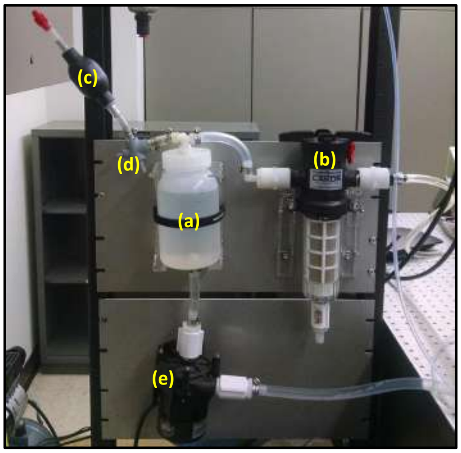


Figure 2.5: Water Pump and Filtration System (a) Pressure Regulation Chamber to capture any air that enters the system and provide a means to remove that air if necessary (b) Water Filter system to ensure tissue slices sucked from the water bath are filtered out before recirculation to avoid obstruction by old tissue during slicing (c) Suction Pump to remove air from chamber using manual suction (d) T-Valve for controlling air inlet and outlet from the chamber (e) Water pump to force continuous circulation of water in the water bath during slicing

This subsystem filters and circulates distilled water in the system. There are four main components in this system - Water Bath, Water Filter, Pressure Regulation Chamber, Water Pump. The components are shown in Fig. 2.5. Care is taken to ensure that the water being used is distilled to prevent rusting of components.

The water bath has one outflow hole, behind the knife, and one inflow hole. The knife is positioned below the water level in order for the system to be closed and complete. The water filter takes water from the suction mechanism incorporated into the knife holder, separates the slices from the water, and then allows the water to move into the pressure regulation chamber. The water filter can be cleaned out easily in this setup as will be required sometimes during tissue slicing. The pressure regulation chamber takes the water from the filter, along with air bubbles that inevitably enter the system. This is to ensure that there is no water in the pump and, consequentially, no clouding of the water bath by the foam that is generated by the pump if air gets in. The T-valve in the pressure regulation chamber is used to suck out extra air from the system. The water pump controls the speed of water flow and can be adjusted for a particular setup using speed levels.

2.3 Improvements on the Initial KESM Prototype

The initial KESM prototype was proven to be capable of imaging several large-scale tissue samples and digitizing them. However, there is a lot of scope for improvements in the design. It was also very expensive and occupied a larger physical footprint.

The IEROM, a second generation prototype based on KESM, is more robust, reliable, and cost-effective. It also reduces the footprint and enables shared use of the instrument via an Internet-connected virtual interface. The new design is one-third the cost and one-fourth the footprint of the original prototype. Software overhaul for the controller part of the instrument has been discussed in Section 2.4, The other improvements to the system are explained in detail in the following subsections.

Optical Subsystem

The new optical subsystems allow flexibility in the choice of objective. This allows us to choose objectives like the Olympus 20X 0.95NA and the Zeiss 63X 1.0NA, with which we can achieve significantly higher resolution. However, for different applications we can change the objective to one of different resolution without having to modify the initial design. The IEROM uses a Dalsa Piranha HS TDI camera, with a responsivity of 1170 DN/(nJ/cm^2), providing approximately a five-fold increase in sensitivity compared to the Dalsa CT-F3 line-scan TDI camera with a 180DN/(nJ/cm^2) responsivity.

IEROM also uses a computer-controlled DC illumination source that reduces the oscillation artifacts introduced by AC voltage fluctuations, which are pronounced due to the speed of cutting. Thorlabs MCWHL2 cold white mounted high power LED is used as the photon

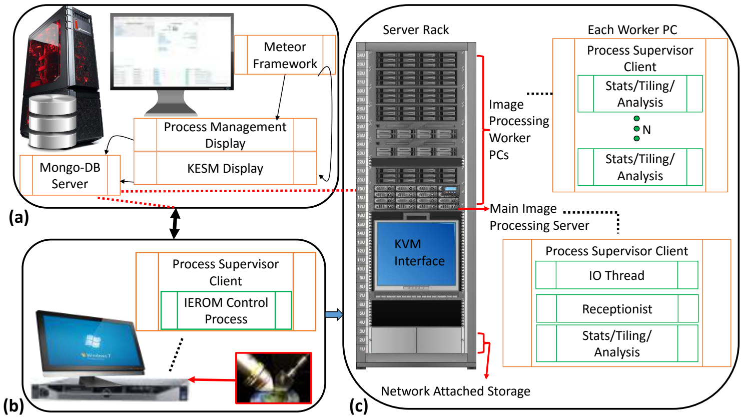


Figure 2.6: Controller Software Overview. *(a) IEROM Instrument Control Module.* The stage controller and the line scan camera of the IEROM prototype are connected to a PC running an IEROM control client. *(b) Management and User Interface Module.* A server hosts the MongoDB database and runs the database process and system display management process for the user interface. *(c) Processing and Storage Module.* A server rack has processing servers to analyze images from the scope for statistics, tile them for display, and compress them before saving. The main processing server receives the images from the microscope and distributes the load to available worker servers. It also has a Keyboard-Video-Mouse (KVM) unit for direct debug if necessary, and houses the NAS to store the digital tissue samples for access in the future.

source and liquid-optic light guide helps in delivering the light to the tip of the knife edge where it is transmitted through the tissue and gets captured by the objective. This is in contrast to the use of laser in the first prototype, which was expensive and also harmful to the eyes, limiting direct observation.

Knife Module

Kinematic principles employed in the new knife design allow for quick swapping of knives without the need for recalibration. The knife is required to have two ports, one for fluid-flow-based tissue removal and the other for light. The fluid flow is achieved using a channel milled between the knife shank and the knife arm, which has a hole leading to the top of the knife arm, which has a tubing connection to the water pump and filtration subsystem. The light port is behind the knife, which can accommodate the fiber optics used to transmit light from the source to the knife. Ball lenses used to be the coupling method for light from the illumination source to the diamond knife. However, this was replaced with a better strategy by coupling the optic fiber directly to the rear-end of the knife module.

Motion Subsystem

The IEROM design has improved reliability and precision of the stage controller motion subsystem by using the Aerotech ANT130-L-Z-PLUS stage with a HPe controller. The linear motion stage itself is a contact-less direct drive technology that uses air bearing, achieving high precision in motion. Modern linear motors with high-precision feedback controllers and low hysteresis minimize long-term error. To add to this, the P̈LUSöption of the stages provide for higher accuracy in the 1V_{p-p} sinusoidal output of the linear encoder. This ensures consistent scanning by enabling higher quadrature output resolution on the HPe controller. Moreover, the instrument also has a custom control board, which transmits location and triggering information between the stage and the camera. This function enables sub-pixel registration of the images in the IEROM.

Water Pump and Filtration Subsystem

In the initial prototype, the water bath used a thick aluminium plate for the base, supported by an angle bracket to attach to the stage. Both the specimen tray and the Y-axis stage rested atop the lift stage, and nearly exceeded its load limit. The bath inlet was provided by a tube epoxy welded to a hole in the side of the water bath. The water bath was opaque and rectangular, making it hard to observe the sample and also wasting a lot of space in the design. The rectangular shape meant more water and thus more load, which worsened the problem of nearly exceeding the lift stage load limit.

The new design solves many of these problems by making the water bath both transparent and an isosceles trapezoid shape to maximize space utilization and reducing the weight of the overall load.

2.4 IEROM Controller Software

The interconnection of servers and components used in the IEROM as well as the processes that help the imaging system work to seamlessly digitize images is shown in Fig. 2.6. In the following subsections, we explain the components and the function of the software processes that run on them.

IEROM Instrument Control Module

This module is responsible for the seamless operation of the automated microscope. It also delivers the data and meta-data from the imaging process to the servers and database respectively. It consists of a PC, microscope automation system, and camera. The PC is connected to the controller, the camera, and the microscope, and runs the process supervisor client which runs the IEROM client as a thread to control the microscope and obtain the images. The PC is connected to the controller and executes the software that drives the stage via the controller. It is also the location of the frame-grabber and is connected to the line scan camera via a high-speed Camera Link Interface. The images from the camera are transferred to the frame-grabber using this interface. The digitized images, however, are not stored in this PC. They are transmitted to the main image processing server using a socket library called ZeroMQ [33] running a protocol called XACT [24]. This involves the controller and camera drivers interfacing to the software framework successfully. The PC also has to manage the buffer of imaged tissue slices and transmission of the data to the other server.

Management and User Interface Module

This module consists of a database server and a web server. These two servers may run on the same workstation or a different workstation, depending on the configuration of the research lab. This server and database are coupled to ensure that any changes made to the database by the control client module or other servers will show up in the interface.

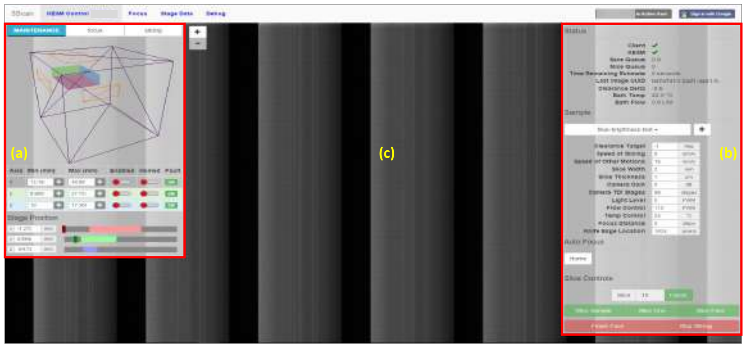


Figure 2.7: Web Interface Overview - Control Page. *(a)* The tissue position indicator and the main panel that lets us select modes of operation and setup the stage for slicing. *(b)* A panel showing system state and sample settings. The sample settings can be changed and initialized during maintenance and the different slicing options can be used to digitize the tissue block in slicing mode. *(c)* The tissue images that have been digitized are shown in the background for the current slice and some previous slices. These are obtained from the tiled images created by servers in the processing and storage module of the IEROM Controller Software.

Similarly, any changes made in the interface will invoke a write operation into the database, which can be used for status change and other control tasks, giving a user control of the machine from a remote web client.

Database System

The IEROM uses MongoDB[12] as the database system. The database is the central synchronization point for all of the involved systems, so it has to be run on a reliable, uninterrupted server in order to handle the large data influx and simultaneously keep track of the process history in all the connected computers. It is the central storage for meta-data related to each image tissue slice and it keeps track of the position of the stage during slicing to revert to the last known position if there is an interruption, without disturbing the registration of the tissue in the digitized image block. It also handles, stores, and tracks records for many parameters associated with the IEROM. This enables authorized users to know and/or change system state. Thus, the system's availability and stability are of utmost importance for reliable operation of the automated microscope.

Web Client and User Interface

The web interface is one of the biggest improvements in the IEROM design. It is based on Meteor [65], a full-stack JavaScript-based platform for developing mobile and web applications. It has multiple pages to give control and monitoring flexibility in many aspects. The administration page shows a list of connected machines and their status. It has a link to a page displaying the processes running on each connected machine and a link to another page that logs all the process messages. Most importantly, it has a link under the Control tab to every instrument that is connected. When it is clicked, the control page of that instrument is shown.

The control page of an automated microscope is shown in Fig. 2.7. This page displays the sample position, stage position, and axes states and stitched tissue samples from previous images in the background. It also shows a setup window with sample setup parameters and the status of the control client process. Any remote user who is logged in can access this page to view the system state. However, the system requires the user to take control of the instrument explicitly. This feature helps to ease shared use of the instrument while protecting it from damage by inexperienced and/or unauthorized users.

Processing and Storage Module

The digitized tissue images received are tiled, analyzed, and compressed before saving them to a predetermined location for further processing. In order to reliably handle the continuous high speed input stream of image data, a group of servers is used alongside a main server.

The main image processing server runs the Receptionist and IO Thread processes that receive the digitized images and process them. The Receptionist is a process that helps balance the workload and perform image data distribution among Worker workstations. Multiple image processing servers can be used as worker nodes to distribute the load of processing the high resolution digital images. This system also houses a Network Attached Storage for storing the images of digitized tissue from the IEROM. Each cluster server runs a process supervisor client that will in turn run the image tile/stats/analysis/compression engines. The tiling is to help visualize the image in series, the compression helps store images in a lesser amount of disk space, and the analyzer displays and stores statistics about the images into the database. Some of these can be viewed on the web interface. To retrieve the images from storage for visualization or processing, one has to access them via the network at the storage location specified in the initial setup process.

Chapter 3

IEROM Installation and User Guide

This chapter is a detailed description on the assembly, setup and operation of the IEROM. Section 3.1 provides an outline of the procedure for preparation and embedding of tissue for KESM imaging. Section 3.2 describes in detail the method of assembling the various components of the IEROM. Finally, Section 3.3 discusses the method and order of setup of the different components for imaging using the IEROM. Thus, this chapter serves as a brief user guide to any user who wants to assemble, setup and operate the IEROM instrument.

3.1 Tissue Preparation

KESM sections tissue block into extremely thin slices of 1μm thickness or lesser. In order to preserve the embedded tissue slice and avoid wrinkling during cutting, a diamond knife is used. However, despite the sharp knife edge, the embedding medium for the specimen needs a certain level of rigidity to preserve the data and avoid damaging the specimen while cutting. Further, if the embedded medium is not rigid enough, then the depending on the order of slicing, the weight of the specimen may shift to one side, causing the whole sample to bend. This will adversely affect registration and alignment in the digitized volume. To avoid these issues, regular embedding resin cannot be used for preparing KESM samples. A special process is followed and care is taken that the tissue sample being imaged is prepared using this process, which is briefly described here.

To prepare the tissue for KESM imaging, it needs to undergo fixing, staining and embedding processes. In this work, only the procedure to obtain a mouse brain stained with India Ink [14] is explained in detail. This will help in understanding how to make the vasculature network in the mouse brain visible for KESM imaging. We use the digitized samples of a mouse brain from an India Ink stained specimen, which was prepared following the procedure described here.

The mouse is first deeply anaesthetised and perfused transcardially with room temperature phosphate buffered saline followed by 10% neutral buffered formalin. This is necessary to clear the blood from the cardiovascular system and to fix the tissues. Then the mouse is perfused with 3.0cc of undiluted India Ink. This is necessary to completely stain the cardiovascular system of the mouse with India Ink. The brain is then removed from the mouse and dehydrated through a series of graded ethyl alcohols. This completes the fixing and staining process for the tissue and it is ready for embedding. Embedding the tissue for KESM can be done in either LR-white or Araldite. Here the araldite embedding process is explained and the LR-white embedding works in a similar way.

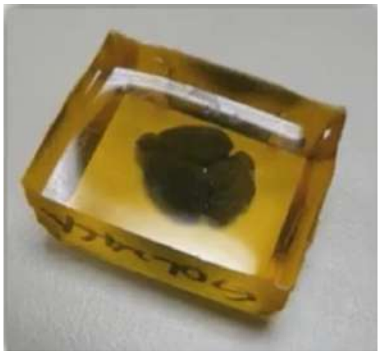


Figure 3.1: A prepared tissue sample embedded for KESM

For embedding the dehydrated mouse brain in araldite, it is transferred to an acetone solution for three to four days, changing to a fresh solution everyday. After this, the brain is left overnight in the refrigerator in a mixture with araldite-to-acetone ratio of 1:2 on the first day, 1:1 on the second day, and 2:1 on the third day. Finally it is refrigerated overnight in a 100% araldite solution for three days, changing to a fresh araldite solution each day.The heated brain is then transferred and embedded in 100% araldite by heating to 60°C in three days. The prepared specimen for a Golgi staining with araldite embedding is shown in Figure

3.1.

Although this section discussed the mouse brain preparation for KESM imaging of vasculature, it can be other tissue specimen like mouse kidney or spinal cord. For staining procedures with Golgi and Nissl stains, the Journal of Visualized experiments has a publication [14] with the detailed description of each of the procedures.

3.2 Instrument Assembly

The main parts of the IEROM system are shown in Figure 2.3. The figure indicates the individual components that make up the IEROM. It is also a reference for the relative placement of all the systems in the assembly being explained below.

Optical Table

The IEROM instrument is assembled on a Newport RS4000-series precision tuned damped top performance optical table. The table rests on four Newport RL2000-series LabLegs^{TM}. The table and support legs together after their assembly should like in Figure 3.2. The table level can be adjusted using a few tools to make it horizontal. This ensures no unbalanced directional load on sensitive instruments such as the positioning stage .

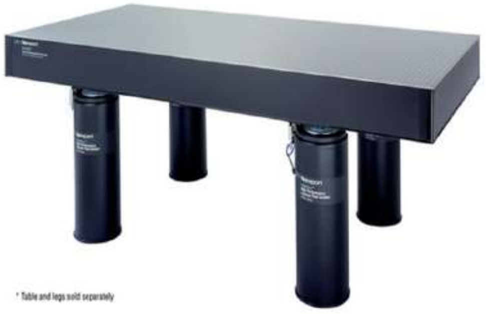


Figure 3.2: AssemblyCheckpoint1: Optical Table from Newport shown with LabLegs^{TM}

Once the table has been aligned and positioned at a convenient location, the load bearing structure is mounted on the table, making sure there is enough place to mount the positioning stage as well. The load bearing structure is two hollow metal pillars connected by a beam. After mounting the structure, its hollow metal pillars are filled will dense and heavy sand for stability. This serves as the backbone and support for the optical subsystem and knife assembly.

Tools for Table Alignment

To adjust the optical table to be aligned and horizontal on the LabLegs^{TM}, a few special tools are needed, namely,

- Regular wrench
- Leveling wrench from Newport for height adjustment of the supporting legs.
- Connectors between the optical table and the supporting legs.
- A high sensitivity spirit level to check the alignment of the table. In the current version of the instrument, a tubular spirit level is used.

Load bearing structure

The opto-mechanical subsystem in the IEROM consists of the line-scan camera, optics train and the knife holder module. These components are supported by a large stress bearing structure called the "H-bridge". The H-bridge is a rigid extruded aluminium structure (XT95 construction rails from ThorLabs) that is filled with special dense-sand to dampen vibrations. It is mounted on the optical table using a base plate for each leg (XT95P3 from ThorLabs). Moreover, to support the optics train, a custom built tip/rotate stage is used. The two different views of this supporting stage is shown in Figure 3.4a and Figure 3.4b.

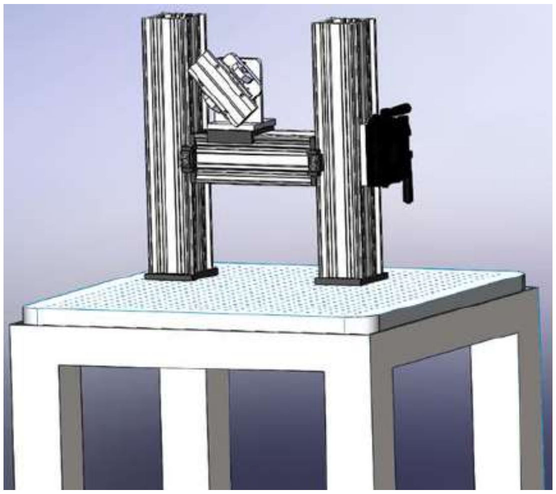
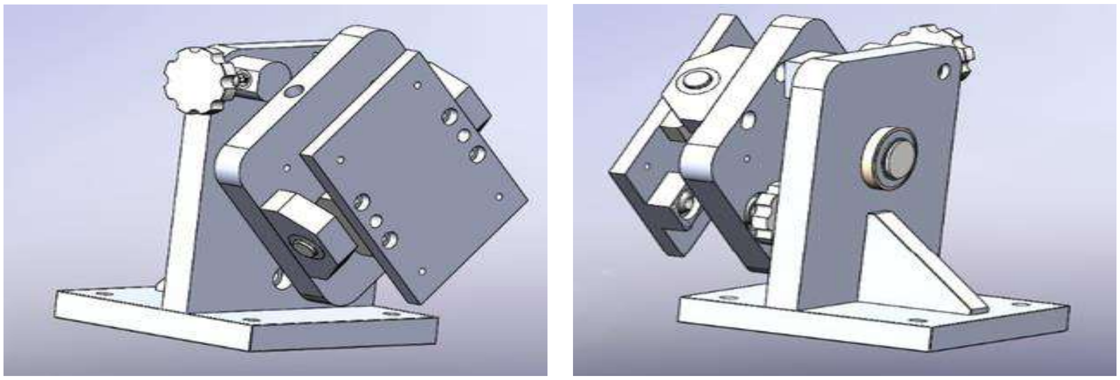


Figure 3.3: AssemblyCheckpoint2: The H-bridge load bearing structure.



(a) Rear view of tip/rotate stage

(b) Front view of tip/rotate stage

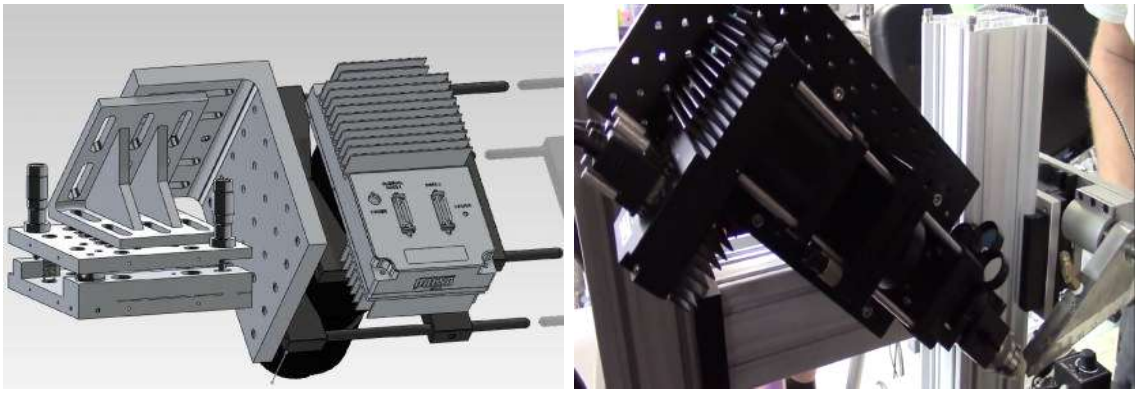
Figure 3.4: Tip/Rotate Stage Models shown from different perspectives

The tip/rotate stage is attached onto the H-bridge structure to support the optics train. Crossed roller-bearing stages are attached to one of the construction rails (the right rail when viewed form the direction in which stage is fixed). This is to attach the knife holder module to the H-bridge structure and to adjust its position and alignment as required. Finally, once the load bearing stage structure has been fixed on the optical table and the movement fixtures described above have been attached to the H-bridge, the system should look as shown in Figure 3.3.

Optics Train and Knife Holder

In this part, the camera, optic train and knife holder are assembled. Although the objective and diamond knife form a part of this subsystem, their installation will be explained later in the process, since they are assembled at the end. To install this system, we need a tip/rotate stage that was assembled as described in subsection 3.2.

First, the optic train along with the camera module is fixed to the left pillar (looking from the side the stage will be fixed on) at the required height and angle. The knife module is placed on the right pillar from the same view. The knife module's position is adjustable as well. To mount the optics train and knife module, use the tip/rotate stage and angle bracket to place it in plane with the knife stage. The rear view of the camera assembly is shown in



(a) Tip/tilt stage used to affix the optic train and camera to the load bearing structure.

(b) The knife holder and optic train have been fixed to the stage.

Figure 3.5: Assembly Checkpoint 3: Setup of Optomechanical parts

Figure 3.5a.

Positioning Subsystem

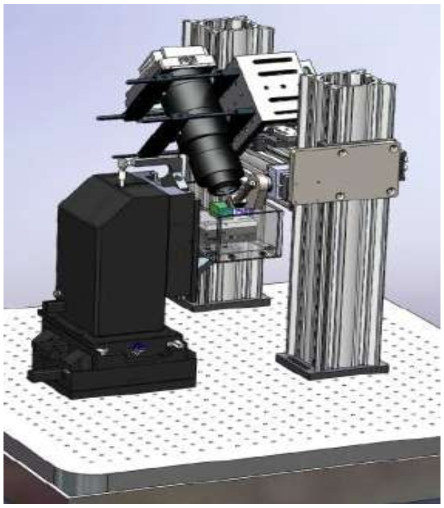


Figure 3.6: Assembly Checkpoint 4 : The stage is also fixed on the base along with the water bath as load.

The positioning subsystem consists of the Aerotech ANT-130-L-Z stage for lift and the Aerotech ANT-130-L-XY stage for motion in the x and y directions. The lift stage rests on the XY stage and the whole system is connected to three controllers, each for one axis of motion. The individual connections are for control and feedback, and the main axis encoder reads the states of the other controllers via daisy chaining.

The plus option version of the controllers is used that can produce higher accuracy of the quadrature signal from the encoder. This high accuracy quadrature signal is

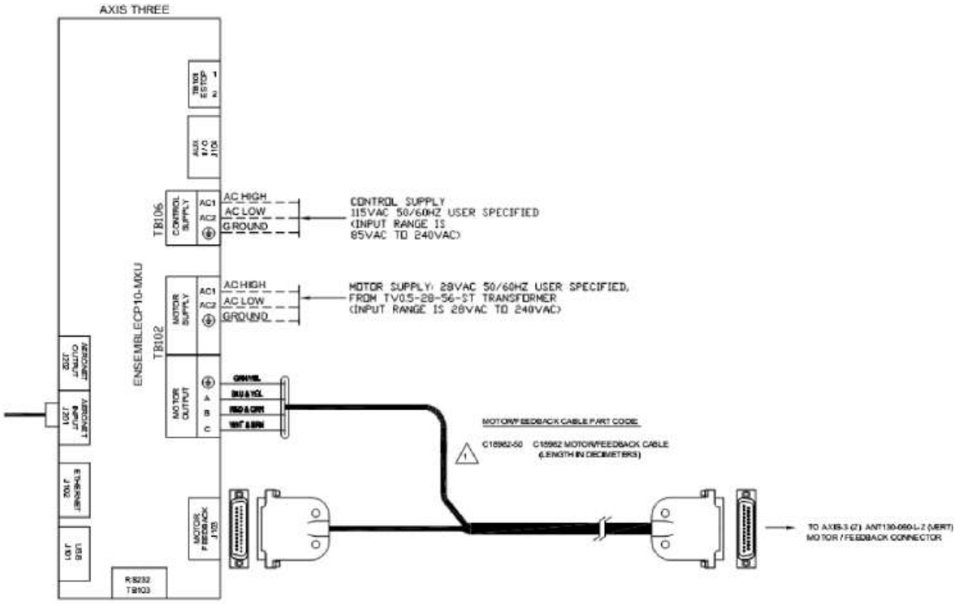


Figure 3.7: A wiring diagram to indicate the connections of the motor of feedback cable from the stage to the Ensemble controller

taken out to a custom design board used to synchronize it with the camera trigger signal, thereby achieving sub-pixel accuracy. The precision positioning stage is fixed atop the table at a distance that will allow for slicing of the tissue in the load but ensure the movement is not restricted because of it. Care should be taken here as the distance measurements are made based on the unit spaced holes on the stage.The stage needs an adapter panel for fixing on the table due to the difference in measurement units between the two. Figure 3.6 shows the system after the stage is also added to the instrument.

To setup the nano-positioning stages from Aerotech, each of them have to be wired and connected to the controllers. The detailed diagram of the wiring for each axis is shown in Figure 3.7. Note that the wiring cables are already provided in the package with the controller from 3Scan so do not tamper with them to ensure the software behavior is as expected. The controllers are daisy chained along their Aeronet interfaces from the first to the second to the third. Specifically,

- Axis 1 - J202 -> Axis 2 - J201

- Axis 2 - J202 -> Axis 3 - J201
- Stage Connector -> Axis 3 (Figure 3.7)

Axis 1 is also connected to the host computer with a USB cord for the controller software to be able to control the system operation. To fix the load onto the stage, the water bath is mounted atop the lift stage using an aluminium right angle support bracket that attaches the bath rigidly to the stages.

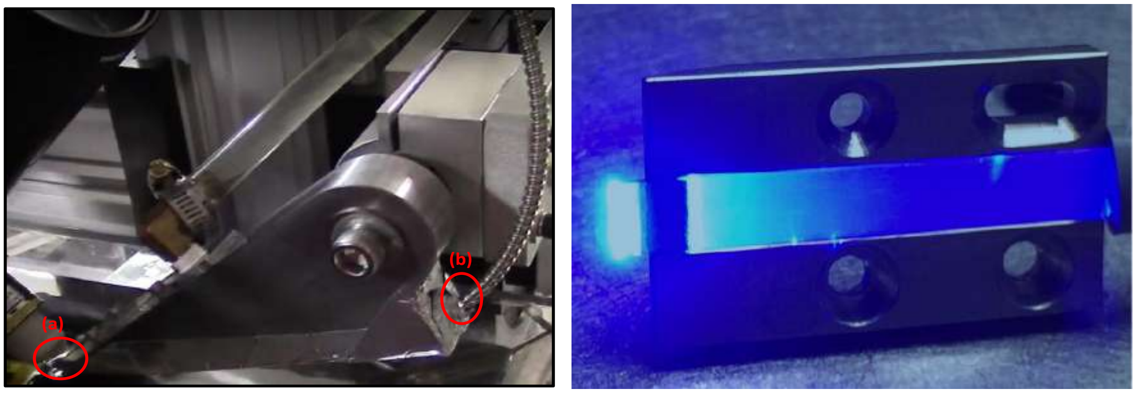
> ⚠ **Check for Sample Holder**
>
> Check if the sample holder is present in the water bath before mounting. If not, you should fix the sample holder to the water bath using the appropriate M6-1.0 screws.

Water Pump and Filtration Subsystem

This subsystem consists of a water filter, a peristaltic pump for circulation, the negative pressure chamber, and the water bath. The water bath is the stage load and has been fixed by this stage in the instrument assembly. To fix the water filter, pump and pressure regulation chamber are fixed as shown in Figure 2.5. Fixing them on such a panel serves two main functions; they are interconnected easily without strain at tubing joints and they can be moved as a single unit if required without re-assembly. The space at the bottom of the panel helps empty the filter out when needed by turning the knob on top, so no other components should be assembled below this in the panel.

Compressed Air supply

The lift stage needs compressed air for operation. So a suitable compressor is chosen to provide a continuous supply of compressed air. An important factor to consider while purchasing the compressor is the noise level in decibels should not exceed 80db. If it does, a noise isolation chamber should be available to place this compressor in. The one being used now, which is a 6.3 Gallon GMC SYCLONE 6310 Ultra Quiet and Oil-Free Air Compressor.



(a) Knife module setup after illumination chain is in place. (a) The illuminated knife tip ready for sectioning and imaging. (b)The light guide is inserted from the rear port of the knife holder and fixed using metal tape after it is coupled to the knife holding kinematic cartridge.

(b) Optic guide coupling to the kinematic cartridge knife holder and transfer of light. Knife edge is bright due to Total Internal Reflection (TIR) and there is no observable leakage before transfer.

Figure 3.8

This compressor is portable as well for flexibility in re-organization. The compressor needs to be regularly cleaned using the water outlet by draining any water that has condensed inside form the moisture absorbed form the the air. If the cleaning is not done, there is a risk of rusting the insides, leading to a very short life span for the compressor.

Apart from the compressor itself, the stage has regulations for the purity of air that can be used to drive it. So a regulator is used that can filter the air before it is given to the compressor. In the Bi2S lab, SMC AWD30-N02DE-Z micro mist separator regulator is used.

Knife, Objective and Illumination Chain

For the illumination chain, an LED light source is used. Thorlabs MCWHL2 cold white mounted high power LED is fixed on the table and is connected to glass fiber optics via an SMT connector. Ensure the fiber optics is placed behind the knife holder connector on the other end. The optic fiber guide can be fixed in place by using the aluminium tape and sealing the rear entry channel in the knife holder that is provided for the guide. The light guide has a locking mechanism to the kinematic coupling for easy setup. The locking mechanism of the optical guide to the knife is shown in Figure 3.8b. The final setup after the optics train showing the light guide entry port, knife position in the knife holder assembly and the exit port of the light at the diamond knife is shown in Figure 3.8a.

Control PC and Servers

Finally, as part of the processing and control system for the instrument, we need multiple PCs and servers that are setup and connected together via the LAN.

A Windows7 PC is connected to the stage controllers via USB interface. The control signals are sent from the software process running on this control PC to the stage controllers which in turn move positioning system to the required position at the set speed. The control software is explained in later sections

A main interface server is used to run the web application using the Meteor framework. This server also houses the database that stores the system state across all devices and connected components. MongoDB is used for database management and its database server will be running on the interface PC. Multiple cluster servers are also setup for distribution of the processing of images that are obtained from the IEROM.

Two fully qualified domain names (FQDN) are needed, one for the web server for the interface PC and the other for the data processing servers with the control PC. This will ensure smooth operation of the remote access feature of IEROM. However, the instrument is built such that it can be operated locally and the remote access is an option. The current configuration uses local LAN network to operate the system.

This was the final step in the assembly process and the system is now all set in terms of hardware and connections. The final system should look as shown in Figure 2.3.

3.3 Instrument Setup for Imaging

In this procedure to get the IEROM ready for imaging, the water filtration and circulation, stage configurations, and sample limits have to be setup.After that, the objective and knife position are tuned for focusing the tissue being sliced.

As a precursor to the setup of instrument for imaging, the control software must be installed in all the computers that have been connected as described previously. The guide to installing the software and setting the correct parameters is available in the lab documents and a user should refer to these manuals and setup the software before following the steps explained in this section.

For imaging with the IEROM, the main tasks include setting up of the water circulation system and to home the positioning stage on all axes. The procedures for both of the tasks are described below in the order they should be carried out.

Water Circulation and Filtration Setup

For the water system, make sure all the connections are secure and the pipes are fastened well at the ends to prevent leakage. Fill the system with water at the water-bath and in the pressure regulation chamber, up to three-fourths the total capacity in both cases.Always use distilled water in IEROM. Using any other water will case corrosion of internal parts and degrade the quality and lifetime of instrument. Moreover, the tissue sample holder is affixed with corrosion-resistant screws and these may rust faster and create debris that might occlude tissue while imaging if regular water is used without distillation.

Next, we should setup the water circulation and filtration. The water bath has one outflow hole behind the knife, and one inflow hole at the far end of the y axis. Thus during the setup, it is important to place the knife below the water level to ensure the completeness of the water circulation loop. Ensure the water level in the negative air pressure chamber of the system shown in Figure 2.5 is upto three quarters of the total height of the chamber. Start the pump/motor in the lowest setting. This will start circulating the water and all the air trapped in the system will start getting trapped in this chamber. If the air gets into the pump, it causes foam in the water bath, which clouds the optics. The pressure regulation chamber allows the water and the bubbles to separate, but extracts the air from the system, creating a low air pressure in the system and regulating the amount of water in the water bath. Once air has displaced more than half the water in the pressure regulation chamber, turn off the pump, open the t-valve on the left of the chamber and suck out some of the water using the manual suction pump. While the motor is on, if the user tries to take air out of the system, it will result in more work and less efficiency. This will not break the motor but it just will not work in high speed settings. It is recommended to either switch off the motor or put it in the lowest setting while trying to remove the air out of the system using manual suction from the pressure chamber.

If the knife is under water but the pressure chamber keeps getting air from the system, switch off the motor check for leaks in the system. Tighten all the joints and tube end points. Lastly, check for the lid of the pressure chamber. If the place of tube joint is not properly seal, use a sealant and fix the leaks. Once the user is sure of an air-tight system, ensure the knife is below water and start the motor again. Repeat the same procedure for taking air out of the system. After around five to ten iterations of removing air, the water level should stabilize and no additional air should get into the pressure regulation chamber. Observe for five minutes that the water level in the pressure regulation chamber is stable. After this, the water filtration system can be assumed as up and running.

Points to Remember

- Always use Distilled water in IEROM water filtration system
- Make sure the knife is under water and the system is leak-proof
- Do not use manual suction with motor at high flow settings. Either switch off the motor or put it in the lowest setting for efficient air removal from the system.

Precision Positioning Stage

The stage needs compressed air to function. For this reason, a compressor in line with a purifier regulator is used whose output is set to around 80psi. Once the water system is setup, turn on the compressor. Adjust the pressure using the regulator-purifier and setup the lift stage. This initial stage setup is required to wait for the water to be filled to ensure the stage is setup for the correct amount of weight as load.

Careful while turning on the Compressor

If the compressor is turned on when the knife, objective and load are fixed, then hold the z-axis of the stage before turning it on so it does not go up with a lot of force and damage these parts. They will be very expensive to replace.

The stage limits need to be calibrated by homing the stage. For this, one first needs to open the control page of the instrument.

Accessing the Instrument Control Page

The instrument control page of the IEROM can be accessed via the local network computer that is running the web servers for the Meteor application. Let this computer's IP address be abc.xyz.pqr.klm It is assumed that both the admin and instrument web application servers have been started. Assume the admin server runs on port 3000 and the instrument server runs on port 3100. Then to access the control page, there are two ways.

In the first method, one directly accesses the instrument's web application by typing abc.xyz.pqr.klm:3100 in the web browser. In the second method, one can go to the admin page by using abc.xyz.pqr.klm:3000 and go to the KESM Control menu option on the top left navigation pane. A drop-down menu lists all the instruments that are currently running and connected to the same database that is bound with the Meteor instance that is running the server interface web applications. Click on the instrument of choice and it goes to the instrument control page. The instrument control page looks as shown in Figure 2.7.

Homing the Stage

All of the measurements along the KESM range from 0-60. To use the scope log into the online interface via Google OAuth API. After you **take control** of the scope you have access to the controls. Make sure the instrument is in the maintenance mode, by seeing if that is the highlighted tab on the left side of the control window between maintenance, focus and slicing.

Do not home with sample in the bath

Homing the stand should only be done when the pump is off, and there is no sample in the bath. If the sample is placed in the bath while imaging, either the knife or the 3D printed knife module cover plate will be damaged. Although the latter is replaceable with a little effort, the knife is extremely expensive and will affect imaging very adversely if damaged.

Now, to enable the stage, all the three axes have to be enabled first by using the respective sliding buttons. Then, home the axes one at a time, by clicking on the sliding button for home corresponding to that axis. Once all the axes are homed, all the sliders should be green. This indicates that the homing process is completed.

Ensure no additional stress on the system

In case of any issues during homing, like a constant ringing noise, ensure that there is no stress on the system and that the controller parameters are setup correctly according to the way it is used in the software. Common causes for unwanted stress on the system are taut wires and connections that pull on the stage during motion, and even if there is friction between the wires and any surface they lie against.

Sample Setup for Slicing

To setup the sample for slicing, the user should specify the sample size by setting limits on the control interface for the slicing motion to start and stop at the correct positions and for

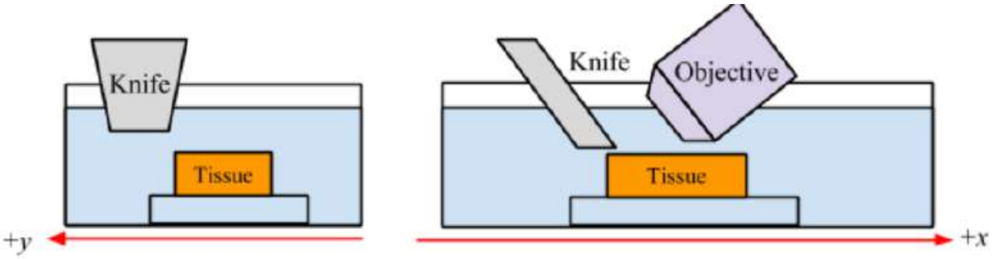


Figure 3.10: Different Views of the Sample as operator moves. *(left)* Front view of the sample as the operator is in Reference Position. *(right)* View of the sample when operator walks to the right side of IEROM.

the slicing to be seamlessly automated.

Assemble the small expensive parts

Before setting up sample limits, the tissue sample is first placed inside the water bath. Then, the objective is placed in its position in the optics train. Finally, the knife is carefully placed in the kinematic cartridge holder. So all the parts are now assembled

Sample Limits Measurement and Setting

To measure the travel limits of the stage, the sample edge measurements have to be noted. The reference position of the operator is shown in Figure 3.9. To determine the minimum and maximum coordinates, the operator must manually move around the stage controller. The different views for x and y limit measurements are illustrated as shown in Figure 3.10.

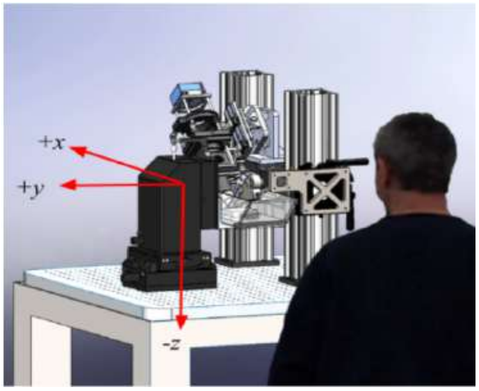


Figure 3.9: Operator position & viewpoint reference.

Careful while setting up sample

- Stage controller axes must be homed before reading the position of coordinates for sample limit setting.
- Before homing, the position of the knife must be placed in a safe location. The knife does not hit the tissue block or the water bath.
- Only one axis that is manually moved must be disabled. Other axes must be enabled not to move alongside the axes.

In this section, the procedure to setup limits in one of the axes is described. Let us choose y axis. Start with the stage controller having x and z axes enabled and y disabled. Move the stage to the farthest position, away from the H-bridge structure such that the knife is away from the tissue sample. The y at this starting position should be maximum. Then move the stage towards the knife module and observe the tissue sample's relative position to the knife without parallax error. The current position is continuously displayed and updated at the left-bottom corner of the tissue position indicator part as S̈tage Position:̈ Once the knife edge reaches the first edge tissue sample, jot down this position as the maximum y. Now keep moving the stage till the other end of the tissue sample is reached by the knife edge. This position is jotted down as the minimum value.

How to set a position without jotting it down

A very handy feature of the control interface is the ability to capture the current position of the stage x, y, or z values without having to note it down.

Refer to Figure 2.7 to locate the features talked about here. To fix a value, click the diamond shaped button near the min or max values of the button to automatically transfer the current position of that particular axis.

Finally, the instrument is now ready to image the sample for which the above setup was done.

Part III

Image Processing and Analysis

Chapter 1

Image pre-processing

Chapter 1 listed the various processing steps in the pipeline. A few of the initial steps are called p̈re-processings̈teps and they are used to remove any unwanted noise or artifacts from images. The data-set obtained from the automated microscope cannot be directly used for visualization or feature extraction. The images contain background noise and additional artifacts specific to the KESM imaging method that need to be alleviated before we can clearly visualize and/or process them.

In this chapter we discuss the various artifacts in the dataset generated by KESM imaging in Section 1.1. Next, Section 1.2 emphasizes the need for the pre-processing steps by illustrating the implications of not having them in different applications. After that, Section 1.4 describes the novel tissue extraction algorithm proposed and published as part of the work towards this thesis. It also discusses in detail the implementation of this algorithm. Section 1.5 briefly discusses the algorithm and implementation of the selective normalization technique [38] to eliminate background noise and artifacts from the images. Then, Section 1.6 concludes the chapter by describing the validation methods used as well as the results of the algorithms discussed in the chapter.

1.1 Image Artifacts in Knife Edge Scanning Microscopy

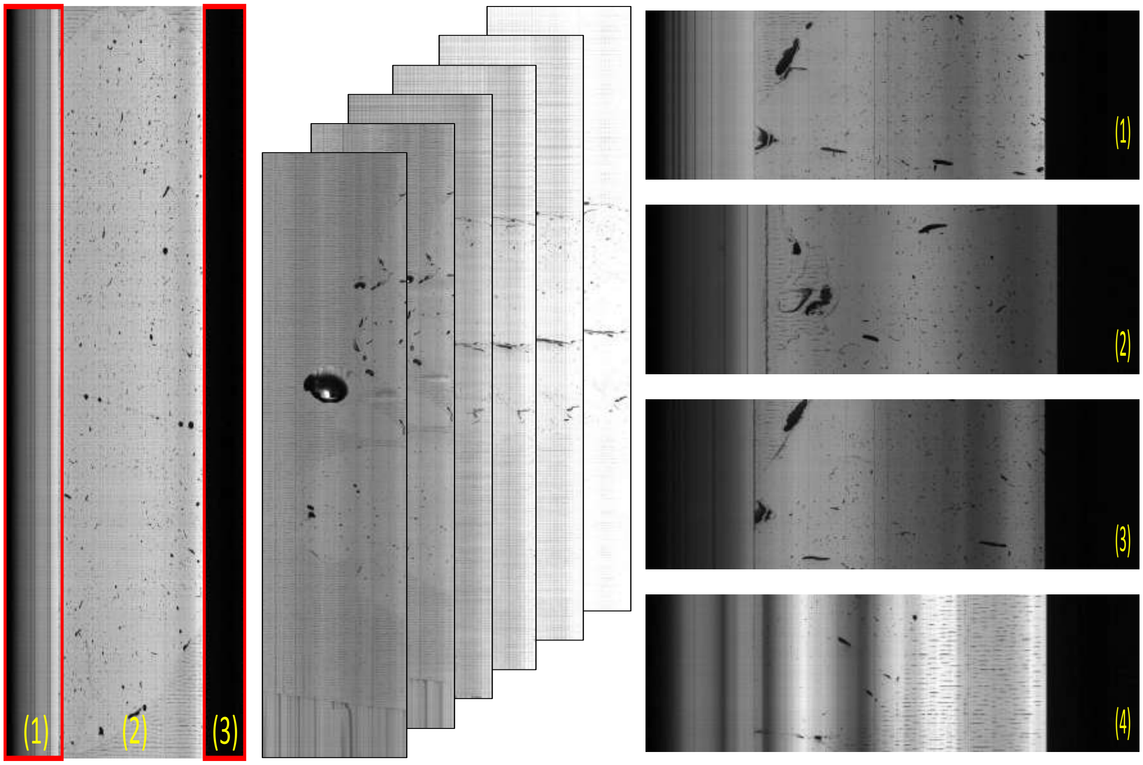
The KESM imaging technique offers multiple advantages over manual sectioning and imaging and other forms of data acquisition. The simultaneous sectioning and imaging approach has following benefits.

- No image registration is required.
- No manual job is required for tissue sectioning, placement or microscope imaging.

However, the data-sets suffer from KESM-specific image artifacts and noise, caused by the side-effects of methods used to achieve high resolution high speed registered images. Some of the main unwanted image characteristics, their probable causes, and need for their mitigation are explained in this section.

Unwanted Image Regions

One of the most prominent unwanted artifact of KESM data is the additional non-informative regions that manifest on either side of the tissue in every image slice. This happens because, while sectioning and imaging in KESM, the width of tissue slice is not exactly the same as the field of view of objective. Thus, there is additional non-tissue area imaged, which appears as dark regions on either side of the tissue in every image. This is highlighted in Figure 1.1a. This is a hindrance to proper visualization and reconstruction of the digitized tissue volume from individual slices for 3D visualization. So the extraction of these regions is mandatory before any further processing can be done. The method developed to remove the unwanted edge pixels have been explained in a Section 1.4.



(a) The image consists of additional non=tissue regions near edges.(1)Extra pixels on the left side of tissue.(2)Tissue region.(3)Extra pixels on the right side of tissue

(b) Due to varying velocity of slicing, the average background intensity of the image can be different. Depending on the illumination power and speed of cutting the image can appear light or dark and this variation can hinder proper segmentation.

(c) The dataset also exhibits intra-image intensity differences. The severity can vary depending on the amount of fluctuation of the illumination source. (1) Mild variation of intensity within image. (2,3) Moderate variation that is noticeable.(4) Highly affected image showing severe artifact that manifests as alternate intensity bands.

Figure 1.1: Artifacts

Illumination Artifacts

Regular Illumination artifacts in the KESM manifest as inter-image intensity differences between slices and intra-image intensity differences that cause a gradient-like background in the images. Irregular illumination artifacts manifest in KESM images mainly due to knife chatter. The reasons for appearance of each of these artifacts is explained below.

Knife Chatter

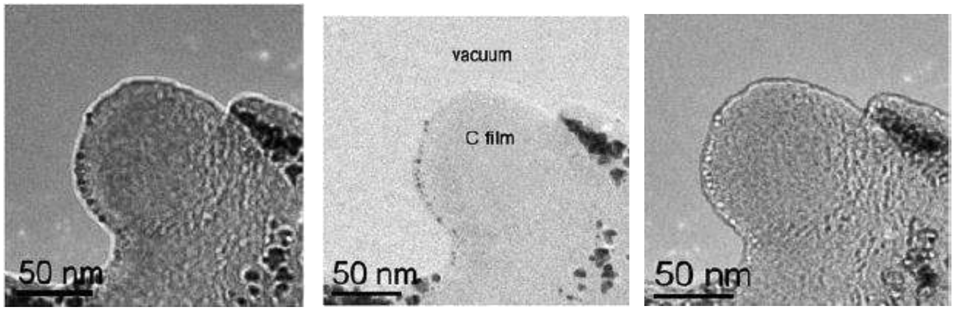
Knife chatter is an issue only when imaging happens simultaneously with sectioning or cutting. This technique is generally only used in machining and hence is well known in that domain. KESM uses this principle of simultaneous sectioning and imaging which causes knife chatter in the images. Knife chatter appears as repeated parallel streaks in the images. A quantitative characterization of knife chatter is difficult since there is no regular pattern with which it appears. Having no regular pattern makes it very challenging to detect the chatter using automated programs.

Intra-image Intensity Difference

This artifact can be described as the changing brightness within the same image. In worst cases this can be seen as vertical band noise or stripes across the width of the image as shown in Figure 1.1b. A main cause for this type of artifact is the variation in illumination power of the light source and the defect is visible along the height of the image slice. The point spread function (PSF) along with such vertical noise produces a grid-noise like effect.

Another reason for this artifact is the presence of any defects on the knife edge causing dark lines along the height of the images in the slices cut with it. This is because the knife itself is the collimator in the KESM imaging technique and any defect on it affects the light transmission resulting in image artifacts. It is present along the vertical axis because of the method of slicing and imaging in KESM.This is not classified as knife chatter so as to maintain consistency with the legacy definition in machining as well as in KESM [45].

Finally, a less frequent but probable cause for this artifact is the misalignment between the objective connected to the camera and the edge surface of the knife. The objective lens is the most important lens within the microscope. It has the strongest influence on the imaging quality. Hence, it is important that the ray bundle runs symmetrically to the optical axis of the objective. In case of misalignment and distortion of ray bundle during initial setup or maintenance, a steady change in overall illumination across the image occurs. An example



(a) Underfocus (b) Focus (c) Overfocus

Figure 1.2: Objective Misalignment causes focus issues. The effects of under focus and overfocus are shown in comparision to the image with correct focus. Adapted from [69].

of how the focus can affect image quality and intensity is given in Figure 1.2.

Inter-image Intensity Difference

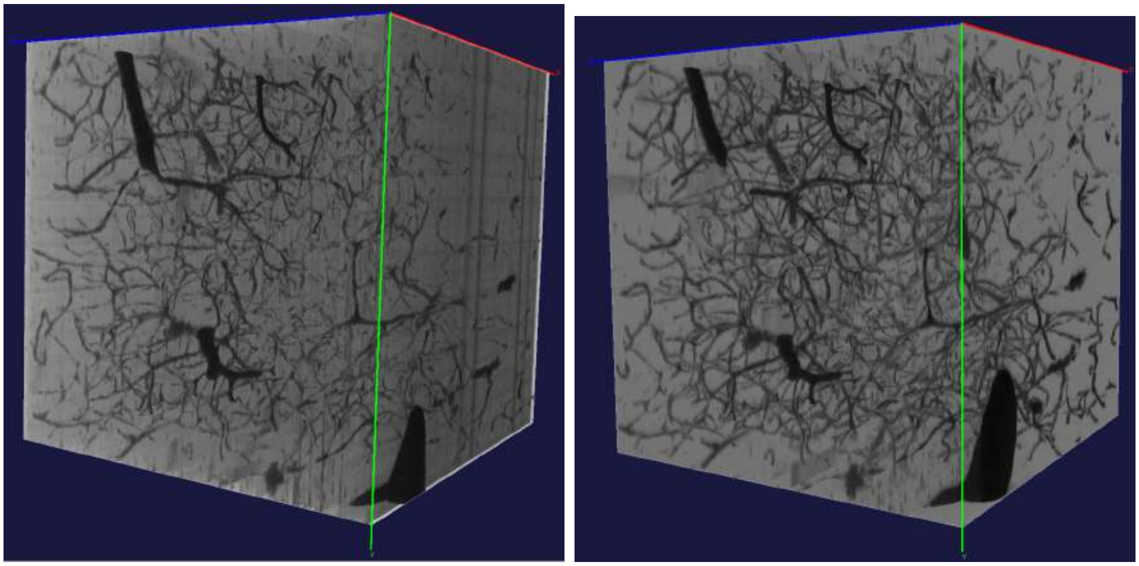
This artifact refers to the variation in background intensity between different KESM images. The cause of this artifact is explained here.

It has been proven that horizontal smudge artifact caused by knife chatter can be reduced by varying the cutting speed [72]. This method was implemented in KESM to reduce the mentioned artifact. Although this helps deal with chatter induced artifacts, the exposure time and hence overall brightness of the image varies each time due to varying cutting speed. Some images are shown cascaded in Figure 1.1c for an estimate of the variation in background intensity that occurs with the variation in cutting speed.

1.2 The Need for Pre-processing

1.3 Visual Quality Implications

The data from the microscope cannot be visualized directly. This is because the images contain all the artifacts that were described earlier in this chapter. These artifacts impede the ability of any visualization tool to distinguish the foreground and background to segment



(a) Raw data. The data quality is compromised due to noise during visualization. Quality of visualization is poor.

(b) Preprocessed data. The removal of noise has made it possible to visualize more information due to better quality of dataset.

Figure 1.3: Visualization of vasculature in a small region of the occipital lobe of a mouse brain.

and show only the structures of interest. An illustration of the difference in the visualization using a tool called Para-View [2], for the same image block with and without pre-processing is shown in Figure 1.3.

Feature Extraction Implications

The pre-processing steps are mandatory for reliable feature extraction and analysis. If the dataset is used as is for feature extraction and analysis most of the artifacts and noise will masquerade as data and features. This is not acceptable, especially in sensitive information like biomedical data that is being published to researchers worldwide. Fea-

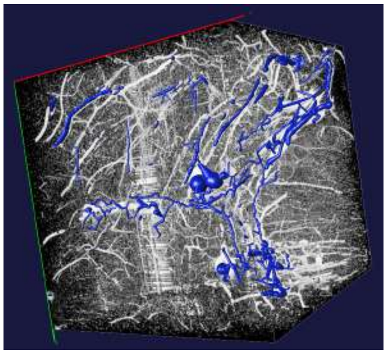


Figure 1.4: Result of feature extraction from a noisy dataset.

tures extracted from the KESM data-sets can be vasculature, neurons and cell bodies. This data is then isolated and analyzed to get statistics from them. These statistics can be used by biomedical researchers worldwide to deduce and infer important structure-function relationships and diagnostic information. This means that there is a need for high reliability and accuracy in the information shared. Thus, the KESM data needs to be clean before processing and analysis and sharing of the important analytical data with the world. Figure 1.4 shows the result of feature extraction from noisy data. The tracing algorithm could not distinguish between the background noise and foreground blood vessel data. This is bad for analytics and most of the statistics derived like volume flow, surface area and even the number of branches might be completely wrong, rendering the processed features irrelevant to the biomedical research community.

Computational Implications

The additional non-tissue area that appears as dark regions on either side of the tissue in every image causes significant increase in the amount of memory required by applications that process the images. KESM could collect terabytes of images and even a small change could make a significant difference in the final result. For instance, assume a tissue sample was imaged using KESM. Let the width of a line scan image be 4096 pixels and a slice consist of 12000 lines. If the images are digitized and stored in the memory using a byte for each pixel, the raw KESM image will occupy 46.875MB. So if we crop the additional region in the image, its width will reduce and thus the memory needed to store it during program execution decreases. In this work, the tissue sample used had tissue width fixed at 2400pixels, which means that only 27.466MB of memory is actually useful. This means that there is a 41.41% reduction in run-time memory requirement for storing the image. This can be used to perform more memory expensive operations to save on time and increase performance of the processing application.

1.4 Automated Tissue Extraction Algorithm

This section describes in detail the algorithm designed to extract tissue regions from the IEROM images by identifying and removing the spurious portions at the edges. The aim is to extract the tissue region in each image of the KESM dataset using this algorithm and also to automate this process so every image is processed and provide a fail safe for an uninterrupted preprocessing pipeline.

Need for a Specialized Tissue Extraction Algorithm

The KESM is designed such that the tissue region should be at a fixed location in the image and the location of the spurious edge for each column is fixed. This would make it possible to crop all images in a given column at the same position, given the tissue width. However, external interruptions and other factors influence this process. More often than not, the tissue location is altered between images, even for those in the same column. Some factors that cause interruptions in the imaging process leading to unregistered images are power outages, clogging in the water circulation, and maintenance issues. While the cause for interruption is fixed and the activity completed, there might be small changes in the objective position, knife position, and sample position, that are now introduced, thereby disturbing registration. So the start of tissue may not be at the same pixel position in the image as before.For these reasons, the cropping at fixed position may cause loss of tissue data. This necessitates detection of the tissue edge for every image.

Each tissue sample imaged by the KESM can generate up to around 80,000 images (the tissue is laterally sectioned several times and each column has around 10,000 images). Thus, to extract tissue region manually requires a lot of time and effort and is inefficient. So image processing is required to automate this.

Existing Algorithms for Pre-processing

The image from the KESM consists of a right and a left edge of tissue as shown in Figure 1.5(a).

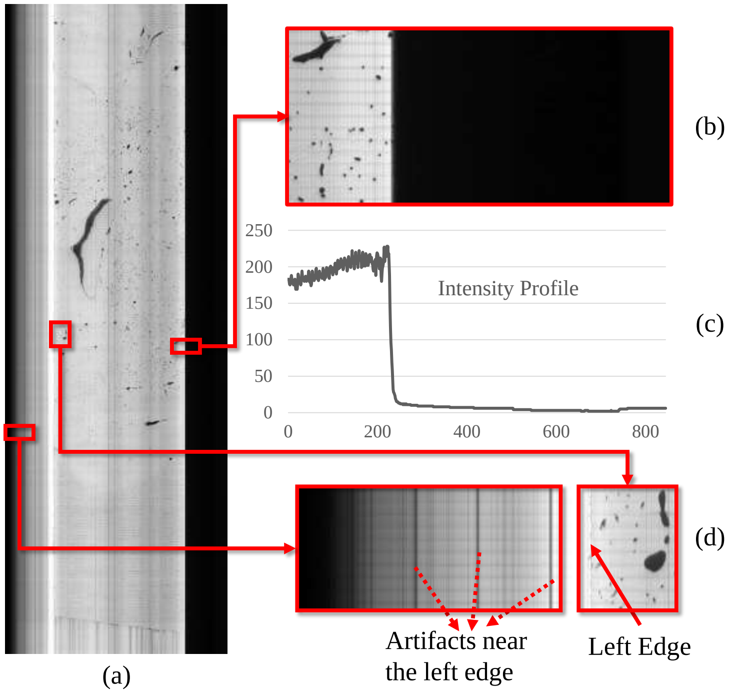


Figure 1.5: Characteristics of edges and spurious edge-pixels in KESM (a) One Tissue Slice from KESM microvasculature data stack. (b) Right Edge of Tissue expanded shows sharp demarcation. (c) Intensity Profile along x axis (distance in pixels from the left side of the image shown in (b)) shows a gradual gradient that might cause spurious edge detection by some edge operators. (d) Left edge (the solid arrow) of tissue expanded shows vertical noise lines (dotted arrows) that have to be distinguished from the real left edge.

Since KESM image characteristics are specific to this imaging technique, not many algorithms exist that target to extract tissue from such images. A template matching based method was proposed for tissue extraction from the KESM image stacks [38]. However, this method suffered from a high error rate. The template matching based algorithm only used right edge template and if it failed to detect the location of tissue edge because of a low template match score or an unclear right edge. This means that a lot of images will either be cropped at the wrong place or fail to crop altogether, causing issues with automation of the data processing pipeline. The novel algorithm for tissue extraction proposed as a part of this thesis has lower error rates and is more efficient in automation of the process.

Novel Tissue Extraction Algorithm

The algorithm for tissue extraction developed in this thesis consists of three logical divisions.

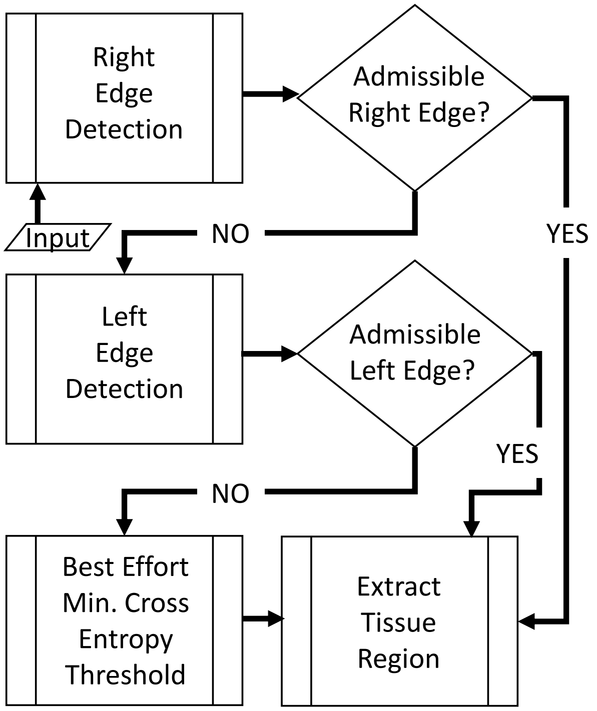


Figure 1.6: Flowchart showing the flow of the novel tissue extraction algorithm

- Tissue Edge Detection: Edge is detected using either the right-edge detection technique, left-edge detection technique or thresholding based edge detection technique explained in this section.
- Admissibility Check: This logical block is designed to run after every edge detection technique applied to check if the tissue area detection was successful or not. If not, it will redirect the program to check the next method of tissue edge detection.

- Cropping: If the admissibility check passed for an edge-detection technique's output, the program corresponding to this logical block is called with the edge location and tissue width of the sample as inputs. Using this information this block will extract the tissue portion of the image and store it for further processing.

The overview of the algorithm flow is shown in Figure 1.6. First part tries to find the right edge of tissue in the KESM image. The output of this part is tested to determine if the right edge found is admissible. If yes, the output is taken as the final index for cropping the KESM image. Otherwise, the algorithm passes control to the second part, which tries to find the left edge of the image. The output of this part is again subjected to an admissibility test. If both these detection methods fail, images are cropped using a minimum cross entropy threshold [41] based edge finder, which is the third part of the algorithm. Note that throughout this section ëdge detectionïs loosely used to mean ẗissue edge detection.̈

Implementation of Tissue Extraction Algorithm

The implementation of the algorithm is elaborated in the sequence shown in 1.6.

Right Edge based Tissue Extraction

Figure 1.5(b) shows the seemingly sharp right-edge. However, Figure 1.5(c) reveals gradual decay in the intensity profile of the right edge. Moreover, KESM is affected by multiple artifacts as described earlier. For this reason, the Sobel [19], Prewitt [61] and similar edge detectors are not used. The Canny edge detector [10] is robust to noise [43] but is complex to compute and gives us additional information about corners that is unnecessary. Hence, the Gaussian gradient is used, which can smooth the image before detecting edge gradients to give us a clear right edge location. The Gaussian Gradient applies a filter in the x and y directions of the two dimensional tissue image being processed. The equations for first order Gaussian derivatives of Gaussian are as given below:

$$\frac{dG(x,y,\sigma)}{dx} = \frac{-x}{2\pi\sigma^4} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{1.1}$$

$$\frac{dG(x,y,\sigma)}{dy} = \frac{-y}{2\pi\sigma^4} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{1.2}$$

,where x and y represent the kernel location starting with zeros at the center and σ represents the standard deviation of the Gaussian distribution. Smoothing effect of the Gaussian Kernel, increases as σ increases.

Algorithm 1 Right Edge Detection

```
function FIND_RIGHT_EDGE
    init
    image ← image * $\frac{dG(x,y,2)}{dx}$
    image ← image * $\frac{dG(x,y,2)}{dy}$
    y,x ← Non-zero Indices from image
    xo ← (val,occ) in x,sorted by freq. of val
    x ← xo for 10 highest ”val” in xo
    index ← val with max(occ) x
    return index
```

The right edge detection algorithm involves finding the right edge at the strongest Gaussian gradient response on the right half of image. The Gaussian gradient function with sigma of 2 is applied. First, the horizontal derivative of Gaussian is convolved with the image (convolution is indicated by $*$) and then the result is convolved with vertical derivative of Gaussian. Ideally, this gives an image with its last non-zero pixel along the x-axis at the right edge of tissue in the image. However, to ensure the algorithm is picking the most well defined edge, the largest non-zero x-index (val) that corresponds most occurrences (occ) of non-zero pixels in y is chosen as the result. The pseudo-code for this procedure is given in Algorithm 1. This works for any KESM image that has sufficient contrast at the right edge. However, if the right edge of tissue is too dark or blurred to recover due to artifacts introduced by the imaging mechanism, this will not give the correct right edge location. The result provided by this right edge detection function is validated by an admissibility test based on tissue width (TissueWidth), sanity checks and image properties near detected edge. If the result is admissible, it is used to crop the tissue from [index-TissueWidth,index]. If not, the left edge detection algorithm, described next, is used to try and find the index of the left edge of tissue region.

Left Edge based Tissue Extraction

Algorithm 2 Left Edge Detection

```
function FIND_LEFT_EDGE
    init
    image ← image $*\ \frac{dG(x,y,1)}{dx}$
    image ← image $*\ \frac{dG(x,y,1)}{dy}$
    image ← Triangle thresholded image
    image ← BinaryErode(image)
    roi_lines ← WIDTH - TissueWidth
    image ← image[0:roi_lines]
    while lines == [ ] and height $\neq 0$ do
        lines ← hough_lines(image,height)
        height ← height - predefined_dec_step
    if lines == [ ] then
        return fail
    index ← x_inx of longest line found
    return index
```

The left edge detection algorithm works by isolating the edge with greater vertical connectivity and also maximum horizontal gradients on the left half of the image. The Gaussian gradient works with lesser smoothing than for right edge detection. This is to ensure the gradient information is not lost while trying to detect the left edge. After this edge detection step, a triangle threshold is used to binarize the image. This image is then given as an input to an optional binary erode filter using a narrow rectangular structuring element. This ensures that only the strongest vertical lines survive. Then probabilistic Hough lines are found in the y-direction and the highest connected vertical line which in turn is the tallest line found is taken as being at the position of the left edge. The x-index at which this line was found is returned to the cropping algorithm. The pseudo-code for this procedure is given in Algorithm 2.

If the left edge found is admissible, tissue region is extracted with the result index value. If not, the image is cropped with a best effort algorithm based on minimum cross entropy threshold [41]. The image is first thresholded based on this method. Then the strongest vertical edge in the image that is farthest to the right is taken as the right edge of the tissue region.

1.5 KESM Image Normalization

After the images are cropped, there is background noise in the image that obstructs the biological structures that are imaged. To avoid this, the images are normalized before feature extraction. However because the artifacts have intensities that overlap or are close to the foreground intensities, segmentation of the foreground structures of interest for further feature extraction and processing becomes very difficult.

Here, we refer to image normalization as the process of making all the images have a predetermined background intensity level. This should also reduce or eliminate knife chatter artifacts and grid noise artifacts. It should ensure that the vertical lines due to knife defects should not cause loss of tissue data in the image. Existing image normalization algorithms cannot guarantee the removal of background noise while preserving the integrity of the biological data present in the image. This necessitates a new algorithm that is specific to KESM data like the one proposed in [38].

This algorithm, which is implemented for normaliztion in this thesis, makes use of equations 1.3, 1.4 and 1.5 to obtain normalized image output.

$$T_x = \frac{p(x,Y)}{M(R(I,Y))} * L \tag{1.3}$$

$$T_y = \frac{p(X,y)}{M(C(I,X))} * L \tag{1.4}$$

$$\bar{I} = \forall_x T_x(\forall_y T_y(I,y),x) \tag{1.5}$$

where I is the output image from of automated cropping, $p(x, y)$ is the value of a pixel at (x, y) in the image, $R(I, Y) = \forall_x p(x, Y)$ is a row, and $C(I, Y) = \forall_y p(X, y)$ is a column in the image. L is the predefined background intensity we are scaling all the images to, $M(list)$ indicates the median value of a list of pixels, T_x is a normalized row, T_y is a normalized column and $\bar{I}$ is the output image from the preprocessing step.

The selective normalization proposed in [38] is also implemented. The threshold value for the median of a row or column was set at 70 and the foreground factor was set at 0.4. These were empirically decided to be the most suitable factors for the KESM Mouse brain vasculature data set that was used to validate the algorithms and data processing pipeline.

1.6 Validation

Results of Tissue Extraction Algorithm

Validation with a real data subset is impractical because finding the correct tissue edge in each image would have to be done manually. This would be subjective, error prone and cannot be considered ground truth. So the proposed algorithm was validated using synthetic images. The algorithm was run on a computer with an AMD-64 quad-core processor running at 3.8GHz with 8GB RAM.

Synthetic Image Generation

Synthetic images are generated by focusing on the edge properties of tissues, the grid noise and vertical line artifacts that manifest in KESM images.

Testing Proposed Algorithm

The tissue extraction code itself that was explained in the implementation section was written in Python. The code for normalization of the data was written in C++. Synthetic images

Algorithm 3 Generate Synthetic Image

```
function RV(inx1,inx2)
    Return random value in [inx1,inx2]
function GenGridNoise(synimg,BKGND)
    VBLOCKS ← #of grids in rows
    HBLOCKS ← #of grid blocks in columns
    var ← random num from 5 to 20
    for i in range(HEIGHT/HBLOCKS) do
        for j in range(WIDTH/VBLOCKS) do
            synimg[i-j-size] ← array[i-j-size]
             using RV(BKGND-var,BKGND)
function GenRightEdge(synimg,BKGND)
    synimg[:,RT:] ← Intensity from BKGND/4
              sub-linearly decaying to zero
function GenArtifacts(synimg,LtEdge)
    n ← random num in [4,12]                    ▷ Num. of lines
    for i in range(n) do
        inx ← RV(0,LtEdge-40)
        widthl ← RV(2,12)
        var ← RV(0,synimg[2,inx])
        synimg[:,inx:inx+width] ← var
function GenLeftEdge(synimg,LT)
    for i in range(HEIGHT/3) do
        synimg[i*3:(i+1)*3,LT-5:LT+5] ←
             array[3x10] using RV(100,150)
        synimg[i*3:(i+1)*3,LT-1:LT+3] ←
             array[3x4] using RV(80,110)
function GenSynImage()
    synimg ← zeros[HEIGHT,WIDTH]
    RtEdge ← random loc in right half of image
    LtEdge ← RtEdge - TissueWidth
    BKGND ← RV(80,150)
    GenGridNoise(synimg,BKGND)
    GenRightEdge(synimg,BKGND)
    GenLeftEdge(synimg,LtEdge)
    GenArtifacts(synimg,LtEdge)
    save(synimg)
```

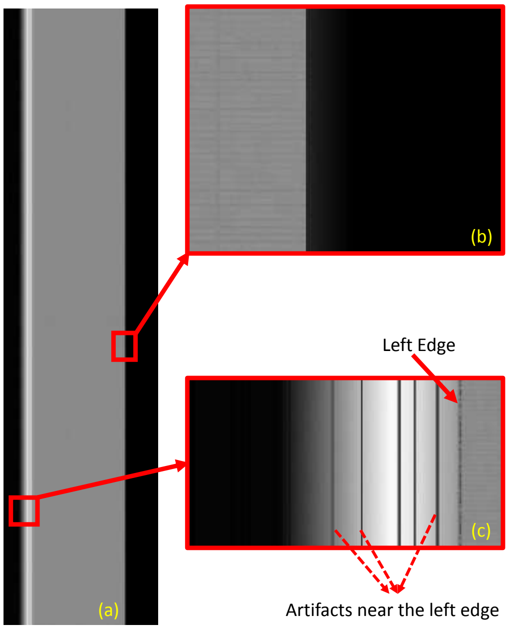


Figure 1.7: Synthetic image. It has edge properties that mimic the real KESM images. These images can thus serve as ground truth for validation of results from the proposed algorithm. (a) Full synthetic image. (b) Right edge magnified. (c) Left edge magnified.

were generated using a python script. They mimic the properties of KESM images and their artifacts. They faithfully reproduce the nature of the edges of tissues in KESM data. An example of an image generated by this algorithm is shown in Figure 1.7(a). The edge properties are shown in Figure 1.7(b) and (c). Thus, we can assume that the results from the synthetic data are directly representative of those from KESM data. We tested the proposed cropping algorithm on hundred synthetic images. From this, the average and maximum errors were found to be 2.68 pixels and 7 pixels respectively.

Comparative Evaluation

We also compared the proposed method to a previously proposed algorithm for tissue extraction. The algorithm being compared is the template matching(TM) based cropping algorithm proposed in [38].

Error in the number of pixels for both these methods is compared in the chart shown in Figure 1.8. From this we infer that the proposed algorithm has improved error rate by 90.92% compared to the template matching based algorithm.The minimum error for the template matching based algorithm for the same set of synthetic images is 15 pixels whereas the proposed algorithm is capable of cropping images with zero error.

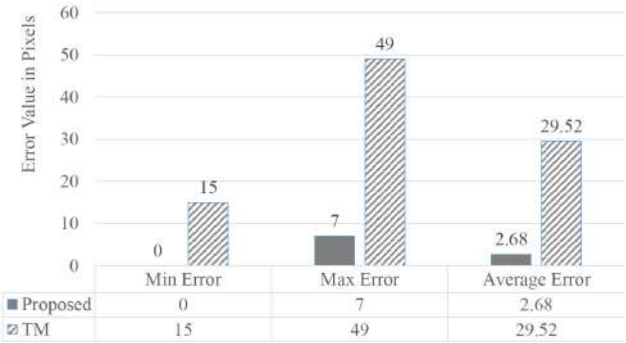


Figure 1.8: Quantitative comparison between the new algorithm for tissue extraction and the template matching algorithm from [38].

Figure 1.9 shows the result of cropping using the two methods on a synthetic image. Clearly, the proposed algorithm has better accuracy in locating the edge of tissue region in the image. Figure 1.9(a) shows a small horizontal section of the synthetic image generated using Algorithm 3. Figure 1.9(b) and Figure 1.9(c) show the result of tissue extraction using template matching algorithm from [38] and the proposed algorithm respectively. Figure 1.9(d) highlights the error from template matching in locating the correct tissue edge (blown up version of the rectangular region from Figure 1.9(b)). This implies that it can extract the tissue region with better accuracy leading to better alignment in the horizontal direction between images.

With the image pre-processing methods applied in this chapter, most of the artifacts that hinder the proper feature extraction from the image data-sets have been removed. The clean vasculature data-set was further processed to trace the blood vessels. The feature extraction is described in Chapter 2. This is the next step of the data processing pipeline that is implemented in this thesis.

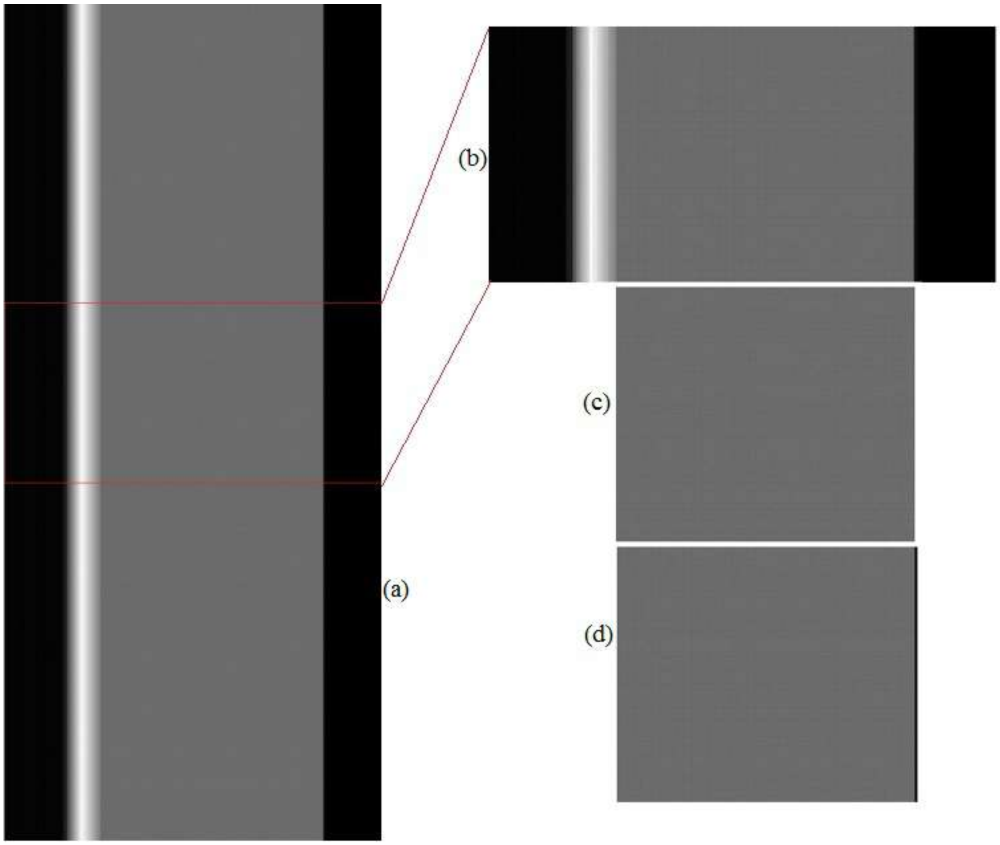


Figure 1.9: Synthetic image. It has edge properties that mimic the real KESM images. These images can thus serve as ground truth for validation of results from the proposed algorithm. (a) Full synthetic image. (b) Right edge magnified. (c) Left edge magnified.

Chapter 2

Tracing Mouse Brain Vasculature Data-set

The clean mouse brain vasculature data is used to extract the important features. The data set is in the form of 2D image stacks which is first converted into sub-divided 3D volume data of size 256pixels×256pixels×256pixels each. The image volumes are then individually traced using an existing tracing algorithm. This chapter reviews the different existing methods for vasculature tracing. It also gives details in the practical implementation of batch processing for the whole mouse brain vasculature data.

There was a proposal made during the course of this thesis to use a divide conquer and combine method to process the whole mouse brain data. However the algorithm developed is still in its infancy and faces multiple challenges. It can be considered as post-processing rather than feature extraction, since the function is technically to stitch together the extracted feature data for visualization, sharing and further analysis. The algorithm and its implementation is described separately in Appendix A.

In this thesis, the volume under consideration is Mouse brain vasculature of the C57L mouse. The vasculature data-set imaged of this mouse using the KESM imaging principle is used for work in this thesis. In this data-set the vasculature data in the images is the main feature of interest. The blood vessels are extracted by tracing the volumes using a known tracing tracing method.

2.1 Review of Existing Vasculature Segmentation Methods

Data analysis often remains the bottleneck in volume reconstruction and progress in research will depend on the computational tools developed to analyze the data [73]. Many algorithms exist to segment vasculature and neurite data from light microscopy image stacks [26, 25, 15, 27] but they all are not generic enough to trace all. Thus, tracing terabytes of data with high accuracy and reasonable speed is a challenge that is still widely researched about today [67, 28].

No known segmentation method has been developed that can segment vasculature morphology from every medical imaging modality. [36]. Existing vasculature segmentation methods can be classified into 3 main categories based on their approach to the problem as follows. [40].

- Appearance and Geometric Models
- Image Features
- Extraction Schemes

In review works a decade ago, the categorization was based on the tracing approaches. For example, the existing methods and algorithms were classified into 6 different categories in [36]. These were were pattern recognition based approaches, model-based approaches, artificial intelligence (AI) based approaches, neural-network (NN) based approaches, tracking based approaches and other tube-like object detection approaches. Some of these categories were further divided internally. Although this review work was detailed, for the purpose of user overview, in this thesis, we use the simpler three category model.

Tools and Components for Extraction of Vasculature

Models

Models are subsets of known properties about the objects in the image to be segmented. These properties can be of two types. One option is based on the appearance of the object in terms of the intensity range it occupies, the image background intensities and nature of noise in the image. or the geometric properties of the object like if it is a tube-like structure or blob-like structure etc.

In case of appearance-based models, the properties are specific to the imaging instrument and need to be tuned every time. This is because the nature of the object's appearance in the image as well as that of the background depends on the method that was used to capture its image. As a very generic and obvious example, we can say that the object's appearance when imaged by light microscopy is very different for dark-field illumination and bright-field illumination. Moreover, the noise pattern introduced in every image is highly dependant on the acquisition method as well as the point spread function (PSF) of the imaging device.

For geometry-based models, as long as the structures imaged are intact, we can segment them accurately using mathematical models of their shape, generally irrespective of the modality. For blood vessels, it is known that they are tubular and elongated, which is valuable information for extraction algorithms to use for blood vessel segmentation. One of the earliest and most prominent algorithms called Multi-scale Vesselness by Frangi [22], segmented vasculature from medical images using this information. Although the approach in that method wasn't purely geometric, it is as an important example for the usefulness of geometric information in vessel segmentation. Frangi's method uses shape-space based on second-order intensity variations. There are also other algorithms similar to this, like 3D MR Angiogram tracing using shape-spaces based on inertia moments [29].

Features

Vessel specific **features** are the practical detectors and filters that are used to evaluate vascular models on the data. One of the category of features are based on local isotropic details like vessel intensity in the image. Some examples of this category of features are ridge based detectors and flux, oriented flux and optimally oriented flux detectors. Other categories of features include 3D local geometry features and 2D cross sectional features. The former includes derivative, integrative, local 3D hybrid model fitting and *a contrario* features. The latter includes cross sectional circular medialness, ray casting features and 2D contour extraction. Other features maybe specifically designed for a particular modality's data properties or to detect certain anomalies like aneurysms and calcification.

Other extraction schemes include region growing, center-line based techniques minimal path detection, active contour schemes like open snake based detection and stochastic frameworks like particle filters and Markov processes.

2.2 Method of Feature Extraction from clean KESM data

Clean KESM data sets can be obtained by following the pre-processing steps described in Chapter 1. Following this, the extraction of tube like structures (features) in the image can be done by applying any of the techniques reviewed previously. In this thesis, blood vessels were extracted from the digitized tissue samples of a India Ink stained Mouse brain. To trace the KESM data-set, the augmented ray-burst sampling [50] method is used for reasonable accuracy at faster speeds. Before describing the details of implementation of batch processing for tracing the mouse brain vasculature data-set used in this thesis, we describe a tool that is used in this process.

Vaa3D and Plugin Use

”Vaa3D” [60, 59] is the short-term for 3D Visualization-Assisted Analysis. It is a tool that is open source and targeted towards medical image processing. It is like ImageJ [1] or Fiji [66] in terms of extensibility. Not only does it provide multiple processing features helpful for biomedical image analysis, it also enables a user to write plug-ins and extend the capability of the tool. The augmented ray-burst sampling based tracing algorithm is one of the programs that has been implemented as a plug-in in this Vaa3D tool. The fact that Vaa3D can be operated using a GUI as well as command line makes it a good candidate for enabling batch processing. Although Vaa3D offers an option to write a plug-in so as to have batch processing built into the plug-in functionality, in this thesis, we use an existing plug-in with a windows batch script for automation.

In the GUI, any plug-in that is developed and integrated to Vaa3D can be seen in the plug-in drop-down menu. In the command line it can be accessed using the following command:

vaa3d -x <unique_sub_string_of_the_plugin_dynamic_library_file>-f <func_name>-i <input image>-o <output>-p <parameter list>-pf <parameter_file>

Windows based automation of the Tracing Process

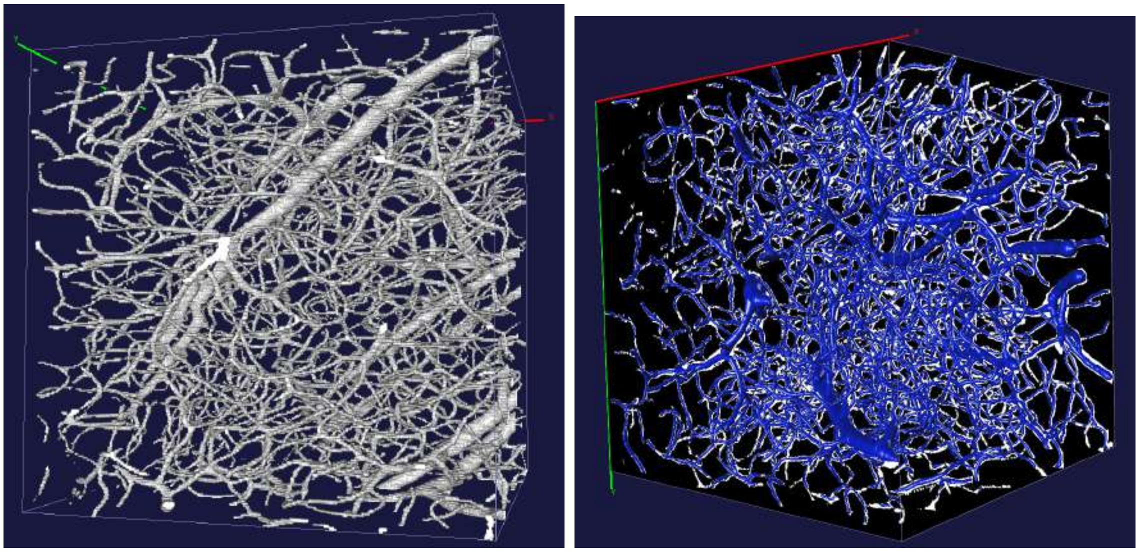
To ensure that the memory limits are not exceeded, the teravoxel volumes are divided further into unit volumes as described before. So the automation process technically involves the tracing of all the sub-volumes one by one (or simultaneous on parallel processors, not implemented in this thesis).

To automate the tracing of volumes a windows batch file (.bat) is written that will call the executable iteratively till all the volumes have been traced.

```
ECHO OFF
E:\
CD "Unit-Volumes"
for %%f in (*.tiff) do (
        vaa3d.exe /x "mostVesselTracer.dll" /f MOST_trace /i %%~nf.tiff
)
ECHO All done tracing.
ECHO Moving all the trace files to separate folder
CD "Traced-Unit-Volumes"

for %%f in (*.swc) do (
        MOVE %%~nf.swc ..\traced-files\
)
ECHO PRESS ANY KEY TO QUIT...
PAUSE
```

2.3 Results of Tracing the mouse brain vasculature

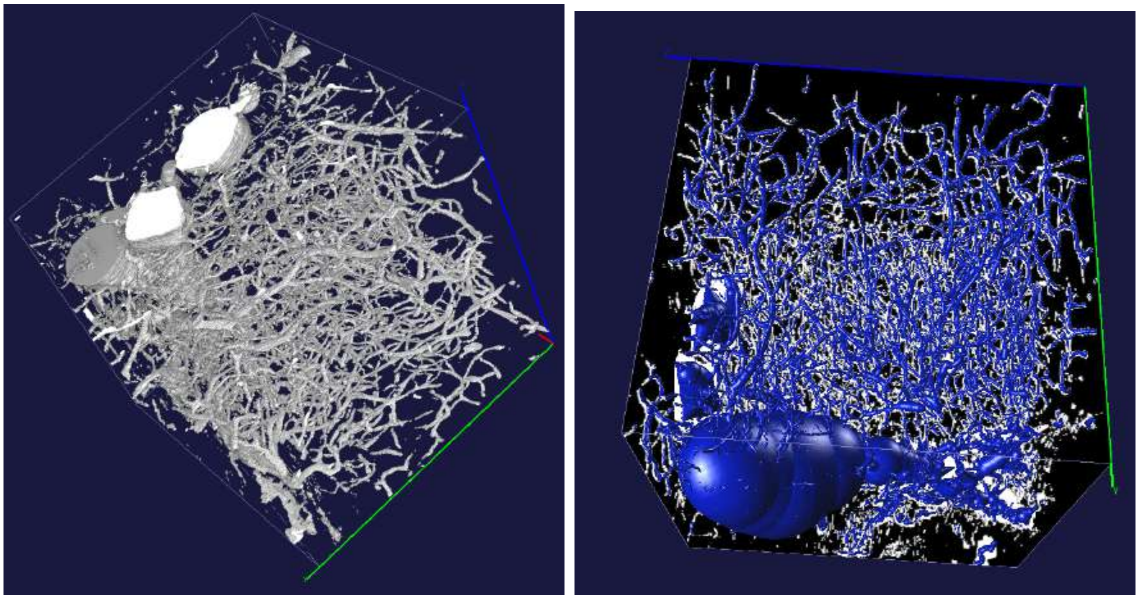
Running the batch file resulted in the processing of all the volume files by using the sub volume files from the input directory which were in TIFF file format [56] as can be inferred from the source code shown in the previous section. Some of these volumes have been visualized using Vaa3D, with and without overlaying the traced results on the original volumes. Two of these volumes are shown in Figure 2.1 and Figure 2.2.



(a) Visualization of clean vasculature data block from the cerebellum.

(b) The traced data overlaid on the original volume shown in Figure 2.1a.

Figure 2.1: Volume-visualization and Tracing shown for a volume size 512×512×512 , extracted from the cerebellum of a mouse brain



(a) Visualization of clean vasculature data block from the Occipital Lobe.

(b) The traced data overlaid on the original volume shown in Figure 2.2a.

Figure 2.2: Volume-visualization and Tracing shown for a volume size 512×512×512, extracted from the occipital lobe of a mouse brain

Chapter 3

Data Representation and Feature Analysis

This chapter deals with the representation and quantitative analysis of the features that are extracted from the mouse brain data set. The tools for quantitative analysis that have been to created and implemented as a part of this thesis are powerful because of the applications of the information that can be derived from them. For instance, disease can be related to structure by quantitative metric comparison between a healthy specimen and an affected specimen.

To get quantitative data from the features, there are multiple novel solutions designed in this thesis. This chapter starts with a description of the file formats that are used in this thesis, to represent the tubular morphology features extracted from the mouse brain dataset. Then, the Tubular Morphology Container (TuMoC) that was fashioned to hold the extracted features in software for analysis is explained. Later, the eXtensible Comprehensive Analysis-engine for Tubular Structures (XCATS) is described. XCATS is the analysis engine planned and implemented in this thesis to extract and store quantitative morphometric information from the features. Finally, the results are shown and an example application has been explained.

First, the Tubular Morphology Container (TuMoC) is proposed as a data container to represent and store the features that have been extracted from mouse brain vasculature volumes. Then,

After extraction of biological structures from the data-set, it is important to analyze the characteristics of the features that are extracted. This can be helpful in medical research to analyze and correlate certain characteristics with diseases. In this thesis, an algorithm has been developed for the extraction of tubular features whose morphology has been described in SWC files [9]. This is done by first designing a data structure that can hold the tubular tree morphology. Then the characteristics of the morphology are calculated as saved in a file. This chapter describes the morphology, the data structure designed to hold it and the calculation of parameters.

3.1 Background

Structure of a Neuron Reconstruction

Reconstructions of a neuron are possible due to feature extraction algorithms that recovered the position information of different points that are part of the neuron called ”nodes” from the original data-set. The characteristics of the neuron at each of these points are also recovered. This is the general method of extracting the neuron morphology to reconstruct the neuronal structure. Nodes, as mentioned earlier, are points that are part of the neuron. The type of node can be determined as one of the characteristics of a neuronal reconstruction and is helpful in visualization as well as analysis. Another characteristic is the radius of the neuronal structure at that particular node, which helps visualize the neuronal structure as a mesh with relative 3D information and also helps estimate the total volume of the neuron. This information helps in neuron modeling for the reconstructed neurons as well. Figure 3.1 illustrates a hypothetical neuron that is reconstructed and visualized using Vaa3D [60, 59]. In the figure, the different colors indicate different type of nodes and the structure can be drawn in 3D due to the position and radius information available from the reconstruction.

Neuron analysis often involves compartmental modelling of dendrites [21], for which tools like Neuron [32] and Genesis [6] are available. However, their input morphology files have requirements that aren't usually met by the reconstructed ones. So a software called CVAPP [3, 9] was designed to handle this by converting an input morphology file format to one that is acceptable as input to the compartmental modelling tools. From the above, it is evident that the methods for reconstruction and modeling of the neuron are quite well established.

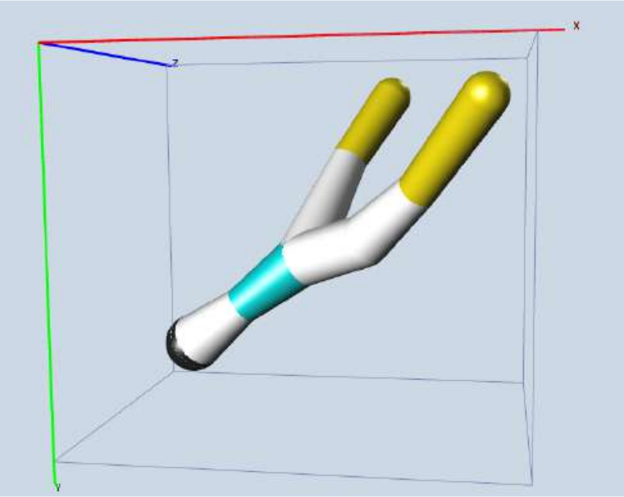


Figure 3.1: An artificial neuron reconstruction visualized using Vaa3D.

Ways to Represent Neuron Reconstructions

The tubular morphologies imaged in the brain are typically neurons and vasculature. Neuron structure has been studied extensively and so there are a few standard available options for representing their structural information. Three popular formats are the Stockley-Wheal-Cannon (SWC) file format, Visualization ToolKit (VTK) Poly-data file format [68] and Neurolucida file format.

The goal in the work of this thesis is to store information about the structure of neurons and vasculature so that we can produce a reconstruction of the Neuron Morphology. The biological structures extracted by feature extraction algorithms as connected structural components (SC). These SCs can be ellipsoid, sphere, cylinder or other 3D topologies depending on the model. SCs are represented as reconstruction nodes each of which have their shape descriptor information. We can describe neuron or vasculature reconstruction as a tree graph. In a reconstruction , graphs are used to describe the topological relationship of all reconstruction nodes. The reconstruction can be stored in different ways and the two used in this thesis are described below.

SWC file format

The Stockley-Wheal-Cannon (SWC) format was proposed by RC Cannon et.al to represent the neuron morphology. It is a simple white-space delimited format to describe a neuron structure. Each line in the SWC describes the properties of a single SC. It consists of the node number(n), node type(t), x-position(x), y-position(y), z-position(z), radius(r) and parent node number(p) in that order. An example swc file for a made-up neuron segment is shown in Table 3.1. The reconstruction of this neuron will look as shown in Figure 3.1.

# n	t	x	y	z	r	p
1	1	14.566132	34.873772	7.857000	0.717830	-1
2	0	16.022520	33.760513	7.047000	0.463378	1
3	5	17.542000	32.604973	6.885001	0.638007	2
4	0	19.163984	32.022469	5.913000	0.602284	3
5	0	20.448090	30.822802	4.860000	0.436025	4
6	6	21.897903	28.881084	3.402000	0.471886	5
7	0	18.461960	30.289471	8.586000	0.447463	3
8	6	19.420759	28.730757	9.558000	0.496217	7

Table 3.1: An example SWC file

Representing Vasculature Reconstruction

Vasculature is also a tubular tree structure, much like the neuron but with some very important differences. More often than not, the vasculature reconstruction and analysis are done using tools that are targeted for neuronal reconstruction, analysis and modeling. Although this works most of the time, it is not tailored to handle the special cases that arise due to differences in the structure of blood vessels and neurons.

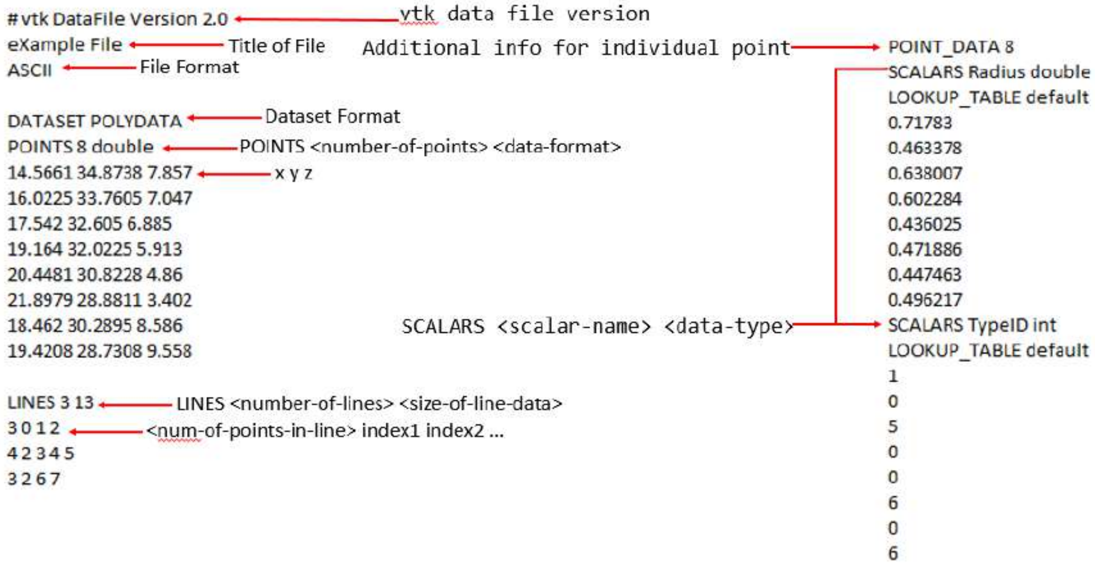
For instance, in many cases of vasculature research, representation of reconstructions is done using swc files that are accepted by tools as neuronal structures with the ”type” characteristic not having any standardized accepted constants for blood vessels. The type field in that file can only represent neuronal component types like axon, dendrite, soma etc. There is no agreed standard yet on what type corresponds to a blood vessel.

Other cases where we can see the lack of proper vasculature specific handling is the feature extraction algorithms. Most tracing algorithms today are tailored for a neuron morphology based processing and feature extraction. Such algorithms will fail to find a soma in the vasculature structure or even insert a soma that isn’t present based on the best fit. In the past few years however, the tracing algorithms are not restricted this way so as to be applicable for vasculature tracing as well. However, there are still features like the high contrast of vasculature data as opposed to neuron data that are not fully leveraged in the algorithms that are designed to trace neurons. Leveraging such features can help with coming up with more efficient algorithms for vasculature.

In this thesis, the data structure used to represent the vasculature data from the reconstruction swc file is described in the next section. However, the format in which the vasculature is stored is converted to a VTK Poly-data format which is very powerful in analysis and . Also, the same format can be used to represent neuronal structures. Scalars can be associated with each point in the VTK Poly-data format. In fact, using Paraview [2] it is possible to visualize the structure and also extract nodes with particular fixed value of scalars or those with scalars in a particular range specified. This can come in handy to visualize just one type of nodes of neuron or to visualize blood vessels only in a specified range of radii.

VTK Poly-data file format

The neuronal or vascular data trace can be considered as connected lines or surfaces. The VTK Poly-data file format is a good way to represent such data. It is defined by the POINTS,



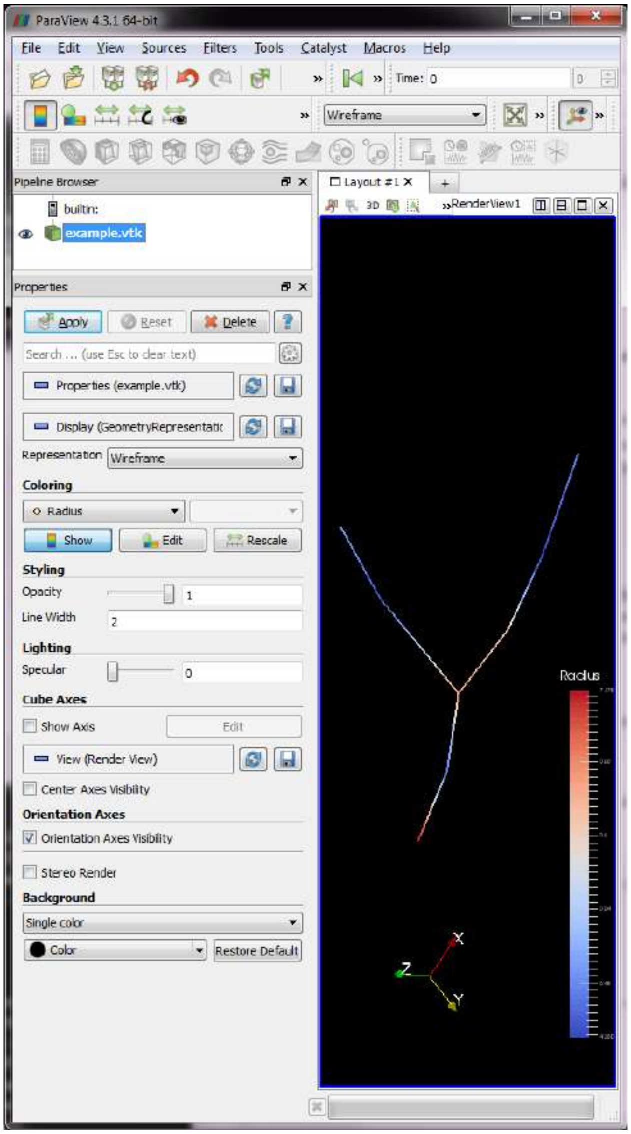
```
#vtk DataFile Version 2.0
eXample File
ASCII

DATASET POLYDATA
POINTS 8 double
14.5661 34.8738 7.857
16.0225 33.7605 7.047
17.542 32.605 6.885
19.164 32.0225 5.913
20.4481 30.8228 4.86
21.8979 28.8811 3.402
18.462 30.2895 8.586
19.4208 28.7308 9.558

LINES 3 13
3 0 1 2
4 2 3 4 5
3 2 6 7
POINT_DATA 8
SCALARS Radius double
LOOKUP_TABLE default
0.71783
0.463378
0.638007
0.602284
0.436025
0.471886
0.447463
0.496217
SCALARS TypeID int
LOOKUP_TABLE default
1
0
5
0
0
6
0
6
```

vtk data file version
Title of File
File Format
Dataset Format
POINTS <number-of-points> <data-format>
x y z
LINES <number-of-lines> <size-of-line-data>
<num-of-points-in-line> index1 index2 ...
Additional info for individual point
SCALARS <scalar-name> <data-type>

Figure 3.2: Example VTK Poly-data File starts on the left continuing on the right.

VERTICES, LINES, POLYGONS or TRIANGLE_STRIPS sections. The file format has extension ".vtk" and supports 1D, 2D, and 3D structured point data-sets. The dimensions must be greater than or equal to 1. The point coordinates are defined by the data in the POINTS section. This consists of x-y-z data values for each point. The VERTICES, LINES, POLYGONS or TRIANGLE_STRIPS keywords define the polygonal data-set topology. Each of these keywords requires two parameters: the number of cells $\boldsymbol{n}$ and the size of the cell list $\boldsymbol{size}$.

For example, if we only wanted to represent the tubular morphology as a set of connected segments, each made up of multiple compartments, each between adjacent reconstruction nodes. To do this, we need the reconstruction nodes as a set of POINTS and segments as set of LINES. Each point can be associated with meta-data like its radius and type, which can be stored as a SCALARS look-up table to access points with specific values of these properties. A simple example VTK Poly-data file shown in Table 3.2. This is the equivalent of the swc file shown in Table 3.1. The VTK Poly-data format representation is viewed in Paraview as shown in Figure 3.3.



Representation: Wireframe

Coloring: Radius

Styling
Opacity: 1
Line Width: 2

Lighting
Specular: 0

Cube Axes
□ Show Axis

□ Center Axes Visibility

Orientation Axes
☑ Orientation Axes Visibility

□ Stereo Render

Background: Single color
● Color

Figure 3.3: Visualization of the VTK Poly-data file in Paraview for the SWC equivalent shown in Figure 3.1. The data is color coded by the radius and the scalar color coding is shown in the visualization window.

3.2 Tubular Morphology Container (TuMoC): A Data Container for Tubular Morphology Reconstructions

Representing the tubular structure reconstruction, be it vasculature or neurons, requires the use of a proper data container. This data container must be representative of the physical structure of the morphology and also provide easy ways to access the data at multiple levels, starting from the node all the way to the entire tree structure. Further the tubular structure reconstruction is referred to simply as reconstruction.

In this thesis work, a Tubular Morphology Container (TuMoC) was designed and implemented in Qt IDE using C++. It was designed in an object oriented approach and is intuitive to understand as a reconstruction. The TuMoC consists of four hierarchically designed classes, using which the reconstruction can be analysed.

- Node Class: This class represents the basic unit of a reconstruction, which is the point of trace that forms the blood vessel. It is associated with type, radius, id and other scalars, that are attributes of this class.

- Compartment Class: This class represents the second level in a reconstruction, which is the morphology between two nodes. It consists of pointers to either end of the compartment.

- Segment Class: This is the highest level of access that is represented as a class, apart from complete tubular tree structure representation itself. It represents the morphology between two special nodes. Special nodes can be of type root, branch or terminal. A segment is a list of compartments that are connected one after another. True to this definition, it is represented as a linked list of pointers to compartments. It also has a list of its parent segments and child segments, that can help traversal of the tube tree structure iteratively.

- TubeTree Class: This class is a representation of the morphology features that have been extracted from the volume. It holds pointers to every hierarchical level of representation. It has a list of root segments that can tell us the total number of separate tree like structures in the given volume. There can be more than one because it could be multiple neurons or because it is a subsections of the vasculature that has two different pathways that might be connected in some other subsection eventually.

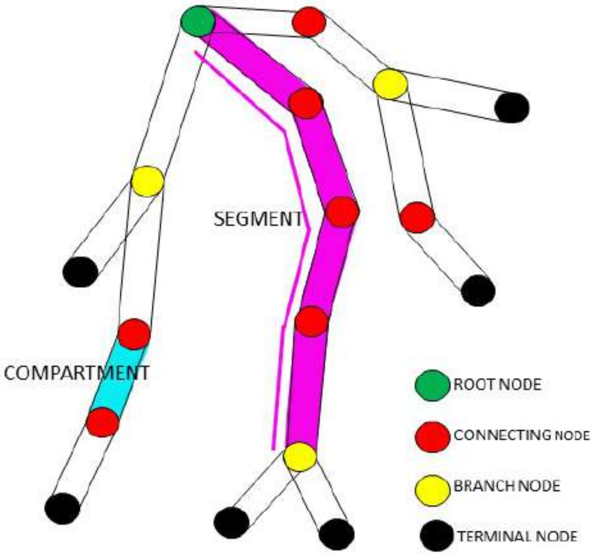


Figure 3.4: Illustration of different components of a tubular structure reconstruction.

Figure 3.4 shows the different components of a reconstruction that have been made accessible in the TuMoC. The nodes are shown as colored circular shapes. The different types of node are shown in different colors and labeled appropriately. The nodes are all given the same radius in this example for simplicity, but in reality these can be different. The designed structure can hold the neuronal reconstructions as well as the vasculature reconstructions in the simple albeit powerful hierarchy described earlier. Currently, this data container provides a backbone to all the processing infrastructure by enabling a common object based access to the tubular morphology that is stored as an swc file after the feature extraction step.

3.3 XCATS: eXtensible Comprehensive Analysis-engine for Tubular Structures

The reconstructed features of the mouse brain can be used for better visualization and analysis, by using the information extracted by the feature-extraction algorithm. However, this is limited to trace-based features like node position, radius, node types, and in some cases, mesh information as well. The need for more morphometric properties to be extracted stems from the need to further analyse and understand the structure of blood vessels. Moreover, comparative studies may need additional information like the number of nodes of a particular type, volume of blood that can flow through a particular region or the average radius information. To calculate all these additional morphometric parameters, and to make them readily available in a web-shareable document format, an eXtensible Comprehensive Analysis-Engine for Tubular Structures (XCATS) was designed in this thesis using the TuMoC as the data container.

In the following subsection, the different parameters that are currently calculated by the engine and the methods used to calculate them are explained in a hierarchical manner.

Parameter Extraction from Node Class

The node is just a point with x, y and z coordinates, its radius r, its ID and parent-ID. There are methods to read each of these properties of the node. The methods are given below.

- getX(): Returns the x-coordinate of the node
- getY(): Returns the y-coordinate of the node
- getZ(): Returns the z-coordinate of the node
- getRadius(): Returns the radius of the node

- getNBifs(): Returns the number of bifurcations originating from the node. If this number if one, it is a continuation node or root node depending on the type. If it is 0, then this is a terminal node. If this number is 2, then it is a bifurcation. This number can be greater than 2 and this is taken by default unless specified otherwise.

- getType(): Returns the type that the node was listed as in the input swc file or the type that was assigned to this node by the feature extraction algorithm.

- getID(): Returns the id of the node in the swc file or the id assigned to the node by the feature extraction algorithm.

- getPID(): Returns the id of the parent node listed in the swc file or the one assigned by the feature extraction algorithm.

Since a node by itself only has these parameters to recover, there are no methods that calculate any additional parameters.

3.4 Parameter Extraction from Compartment Class

The compartment class represents the connection between two nodes. This can be treated like a volume shaped like either a frustum or a cone between two nodes. The parameters for this type of structure that need to be estimated are volume, surface area, length. the different parameters and their calculation is explained below.

- Length: This is the euclidean distance between two points of the compartment. If the two points are (x_1, y_1, z_1) and (x_2, y_2, z_2), then the euclidean distance is calculated as below. The formula used to calculate this parameter is

  $$L = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{3.1}$$

  , where L is the length of the compartment and x_n, y_n, z_n are the x, y and z coordinates in the three directions.

- Cylindrical Volume: This is the volume of the compartment assuming the reconstruction between the nodes is cylindrical with the radius of the second node. The formula used to calculate this parameter is

  $$V = 4 * \pi * (R_2)^2 \tag{3.2}$$

  , where V is the resulting volume, π is a mathematical constant, the ratio of a circle's circumference to its diameter, R_2 is the radius of the compartment's end node.

- Cylindrical Surface Area: This is the lateral surface area of the reconstruction to find out the surface area of the tubular tree. In this parameter calculation we assume the reconstruction is cylindrical. The formula used to calculate this parameter is

  $$S = 2 * \pi * R_2 * L \tag{3.3}$$

  , where S is the resulting surface area, π is a mathematical constant, R_2 is the radius of the compartment's end node, L is the length of the compartment that is calculated as described previously.

- Frustum Volume: This is the volume of the compartment assuming the reconstruction between the nodes is a frustum. This parameter is calculated as follows

  $$V = \frac{\pi}{3} * L * ((R_1)^2 + (R_2)^2 + R_1 * R_2) \tag{3.4}$$

  , where V is the resulting volume, L is the length of the compartment, R_1 is the radius of the first node of the compartment and R_2 is the radius of the second node of the compartment and π is the mathematical constant.

- Frustum Surface Area: This is the lateral surface area of the reconstruction to find out the surface area of the tubular tree. In this parameter calculation we assume the reconstruction is a frustum. The formula used to calculate this parameter is

  $$S = \pi * (R_1 + R_2) * \sqrt{(R_1 + R_2)^2 + L^2} \tag{3.5}$$

, where S is the surface area, R_1 and R_2 are the radii of the two nodes that make up this compartment, L is the length of the compartment and π is a mathematical constant.

3.5 Parameter Extraction from Segment Class

The Segment class represents sequentially connected compartments. The parameters for this type of structure that need to be estimated are volume, surface area, path length and euclidean length. the different parameters and their calculation is explained below.

- Volume: The volume of any given segment is calculated as the sum of volumes of all compartments. Like the compartment, the segment volume can be calculated as sum of either cylindrical or frustum volume of every constituent compartment of that segment, which can be labelled as the cylindrical or frustum volume of that segment. The calculation is however not repeated and the volume is calculated by the sum of volumes of the compartments it contains. This can be evaluated as shown below. Let the number of compartments in the segment be N.

$$V_{cyl} = \sum_{n=1}^{n=N} Vol_{cyl}(Compartment_n) \tag{3.6}$$

$$V_{frstm} = \sum_{n=1}^{n=N} Vol_{frstm}(Compartment_n) \tag{3.7}$$

  , where V_{cyl} is the volume of of the segment assuming compartments are cylindrical, V_frstm is the volume of the segment assuming compartments are frustums and $Compartment_n$ is the n^th compartment of the segment and Vol_{cyl} and Vol_{frstm} are the volume calculation functions of the compartment assuming it is a cylinder and frustum respectively.

- Surface Area: The surface area of any given segment is calculated as the sum of the surface areas of all compartments. This can be evaluated as shown below, where we assume the number of compartments in the segment are N.

$$S_{cyl} = \sum_{n=1}^{n=N} SA_{cyl}(Compartment_n) \tag{3.8}$$

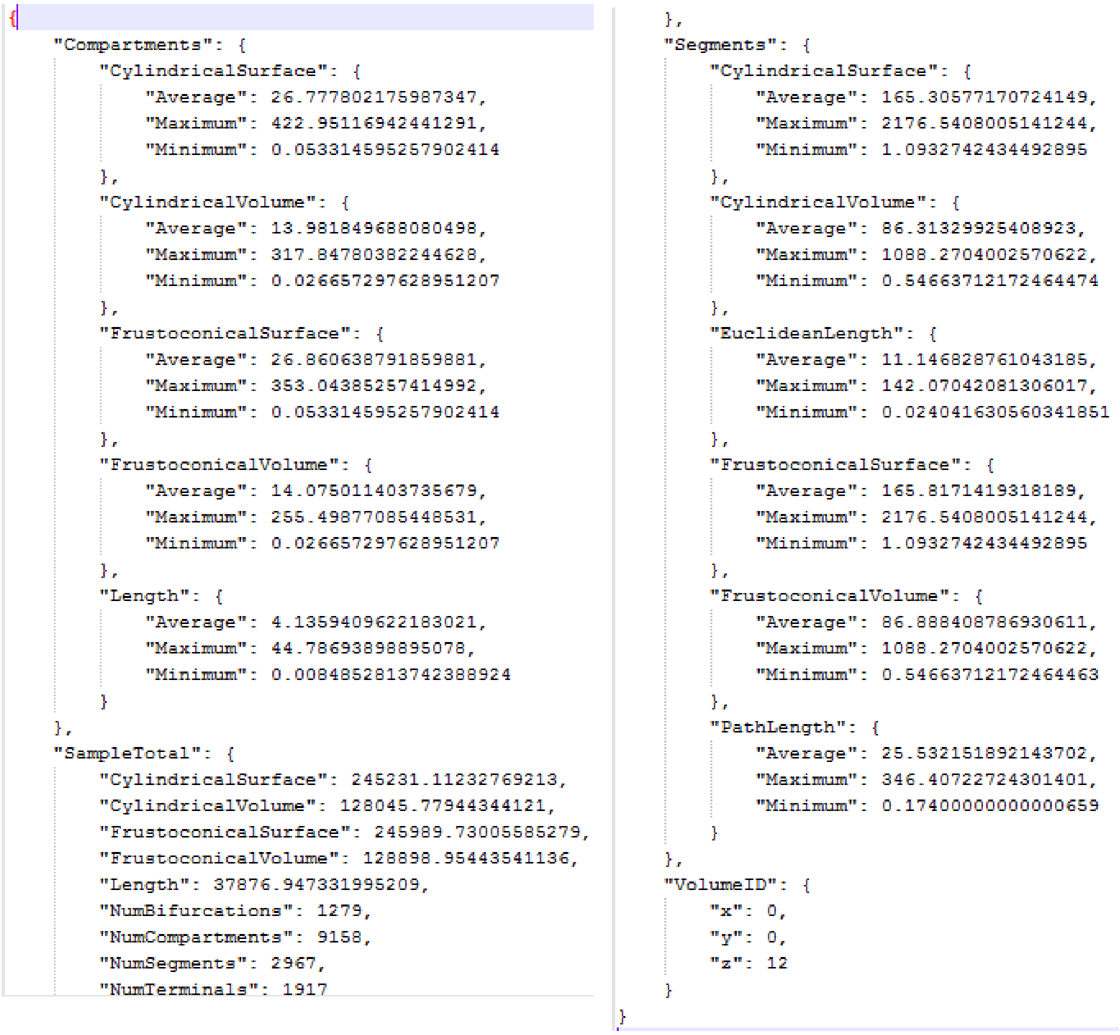
$$S_{frstm} = \sum_{n=1}^{n=N} SA_{frstm}(Compartment_n) \tag{3.9}$$

, where S_{cyl} is the lateral surface area of of the segment assuming compartments are cylindrical, S_frstm is the lateral surface area of the segment assuming compartments are frustums and $Compartment_n$ is the n^th compartment of the segment and SA_{cyl} and SA_{frstm} are the lateral surface area calculation functions of the compartment assuming it is a cylinder and frustum respectively.

Using the analysis engine built in this work, the results from Chapter 2 are analyzed. Statistics from the entire mouse brain vasculature are extracted. The results are stored in a web-shareable document format called JavaScript Object Notation (JSON) format[7]. This format consists of objects and arrays. Either can be nested within the other or itself. The elements that form the arrays or objects are string value pairs. Values can be strings, numbers, objects, arrays, true false or null. These data structures are universal in the sense that virtually all the modern programming language support them. So it can be easily used as an interchange format. It is especially used in web based applications as data exchange formats. An example output JSON file obtained from XCATS for a volume of mouse brain of size 256x256x256 at the highest resolution is shown in Figure 3.5.

3.6 Format Converter from SWC to VTK Polydata

The TuMoC reads the reconstruction data into a convenient structure for conversion, processing and analysis. It can convert these reconstructions into VTK Poly-data format with segments listed as collection of points that are called lines. The radius and type information of each node is contained as a table of scalars. In this implementation, each parameter



```
{
    "Compartments": {
        "CylindricalSurface": {
            "Average": 26.777802175987347,
            "Maximum": 422.9511694244129l,
            "Minimum": 0.053314595257902414
        },
        "CylindricalVolume": {
            "Average": 13.981849688080498,
            "Maximum": 317.84780382244628,
            "Minimum": 0.026657297628951207
        },
        "FrustoconicalSurface": {
            "Average": 26.860638791859881,
            "Maximum": 353.04385257414992,
            "Minimum": 0.053314595257902414
        },
        "FrustoconicalVolume": {
            "Average": 14.075011403735679,
            "Maximum": 255.49877085448531,
            "Minimum": 0.026657297628951207
        },
        "Length": {
            "Average": 4.1359409622183021,
            "Maximum": 44.78693898895078,
            "Minimum": 0.0084852813742388924
        }
    },
    "SampleTotal": {
        "CylindricalSurface": 245231.11232769213,
        "CylindricalVolume": 128045.77944344121,
        "FrustoconicalSurface": 245989.73005585279,
        "FrustoconicalVolume": 128898.95443541136,
        "Length": 37876.947331995209,
        "NumBifurcations": 1279,
        "NumCompartments": 9158,
        "NumSegments": 2967,
        "NumTerminals": 1917
    },
    "Segments": {
        "CylindricalSurface": {
            "Average": 165.30577170724149,
            "Maximum": 2176.5408005141244,
            "Minimum": 1.0932742434492895
        },
        "CylindricalVolume": {
            "Average": 86.31329925408923,
            "Maximum": 1088.2704002570622,
            "Minimum": 0.54663712172464474
        },
        "EuclideanLength": {
            "Average": 11.146828761043185,
            "Maximum": 142.07042081306017,
            "Minimum": 0.024041630560341851
        },
        "FrustoconicalSurface": {
            "Average": 165.8171419318189,
            "Maximum": 2176.5408005141244,
            "Minimum": 1.0932742434492895
        },
        "FrustoconicalVolume": {
            "Average": 86.888408786930611,
            "Maximum": 1088.2704002570622,
            "Minimum": 0.54663712172464463
        },
        "PathLength": {
            "Average": 25.532151892143702,
            "Maximum": 346.40722724301401,
            "Minimum": 0.17400000000000659
        }
    },
    "VolumeID": {
        "x": 0,
        "y": 0,
        "z": 12
    }
}
```

Figure 3.5: An example JSON file showing the different components, with the self-explicable key-value pairs

value associated with each node is written into the VTK file as its own scalar look-up table. This helps in sorting the nodes and extracting nodes with any one of the parameters in the ParaView tool.

This can be extended to represent the structure as a mesh surface that will be helpful for volume visualization. The idea is, rather than representing entire volume as a segmented data-set, displaying a mesh structure of the reconstruction will occupy less memory. If annotated with the meta-data that is available in this structure, giving more insight into the object structure at the same time. Further, it is possible to build extensions to the data container in the form of statistics and storing into other formats for visualization. It can also be used to create better visualization for the tubular reconstructions extracted from medical data sets. When used directly during feature extraction by tracing algorithms, this container can be extended to store additional data information and eventually it can be stored as mesh structures using VTK files. This is a very convenient way to extract data from the imaged data set and store it with just the relevant information to the medical community.

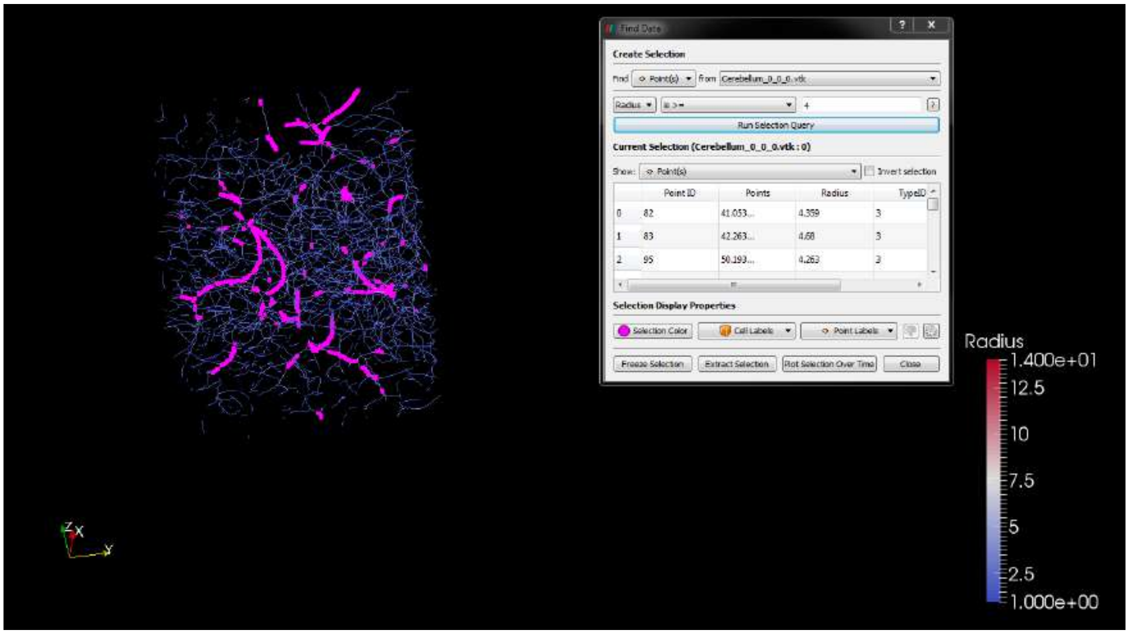
It is notable that the use of this algorithm is not limited to the IEROM data-set. Since it processes any file in the SWC format, it can even accept data from different neuron and vasculature reconstructions in this format and convert it to VTK format. There is also statistical analysis built into the containers that can be invoked to get the value of different parameters that are important to diagnose and analyze the reconstructions. This means that statistics from different neuron and vasculature reconstructions can be analyzed. The statistic calculation part of the container is explained below.

The resultant .vtk file is visualized in Figure 3.6. Saving a .vtk file is advantageous for fast look-up of nodes according to the scalar meta-data associated with them. In Figure 3.6 , the window showing the filter used and the points highlighted in the trace and listed in the window are shown to demonstrate the powerful control over data provided by VTK files.

3.7 Quantitative Analysis Example

To demonstrate the use of XCATS and TuMoC, two regions of the mouse brain were compared quantitaively using the pipeline built so far. A small portion of the cerebellum of size 512×512×512 shown in Figure 2.1 is compared against a region from the occipital lobe of the same size, from the same mouse brain dataset Figure 2.2. The two regions are from the same range of depth in slicing for a fair comparison. The results from this comparison for the branch count, volume, and surface area are shown in Figure 3.7.

There are multiple features that can be added to the analysis engine such as local and



Find Data

Create Selection

Find: Point(s) from: Cerebellum_0_0_0.vtk

Radius: is >= 4

Run Selection Query

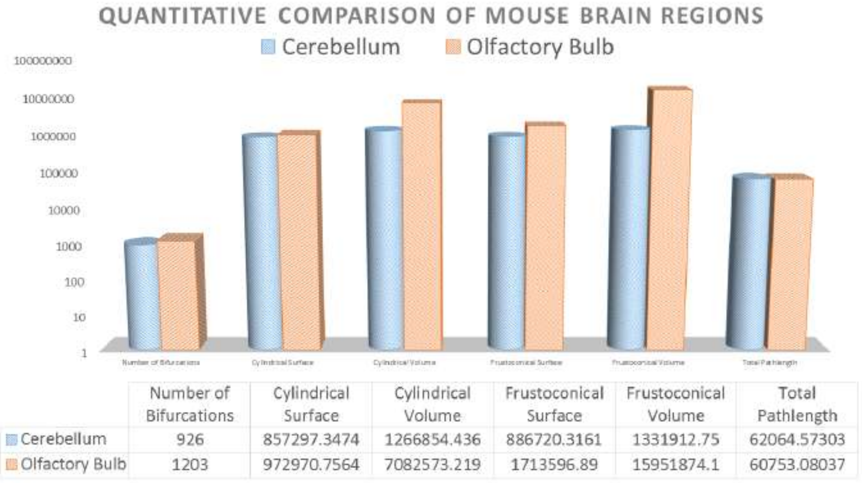
Current Selection (Cerebellum_0_0_0.vtk : 0)

Show: Point(s) □ Invert selection

	Point ID	Points	Radius	TypeID
0	82	41.053...	4.359	3
1	83	42.263...	4.68	3
2	95	50.193...	4.263	3

Selection Display Properties

Figure 3.6: Visualization of the VTK file trace of Cerebellum volume shown in Figure 2.1 using ParaView. A filter of radius greater than or equal to 4 is applied to get the points from the entire trace that satisfy this condition. These are highlighted in the VTK visualization as shown.



	Number of Bifurcations	Cylindrical Surface	Cylindrical Volume	Frustoconical Surface	Frustoconical Volume	Total Pathlength
Cerebellum	926	857297.3474	1266854.436	886720.3161	1331912.75	62064.57303
Olfactory Bulb	1203	972970.7564	7082573.219	1713596.89	15951874.1	60753.08037

Figure 3.7: Comparison of the two regions extracted from a mouse brain for quantitative description and understanding of the biological features. The graph is plotted from data generated by the analysis engine XCATS developed in this work.

remote bifurcation angle of compartments and segments. One can also calculate the branch order of any particular segment, the Hausdorff distance, fractal dimension and bifurcation torque parameters. In fact, this work is extensible in itself, but a slightly different method of implementing the data container in this analysis engine is by separating the parameter calculation methods. However, the choice to keep the current method was made due to the difference in the function implementation for the same parameter between a compartment and a segment. As we have seen, compartments are put together to form segments and segment parameters are calculated as the sum of compartment parameters. But this only holds true for volume and surface area and to implement the extended analysis it will be a better solution to adopt a new class as the parameter calculator in the new container.

Part IV

Conclusion and Future work

Chapter 1

Conclusion and Future Work

Although the resultant data-set from this thesis has all the necessary information it is not shared with the biomedical research community. This needs additional steps and tool development to be incorporated into the data pipeline which will be discussed in Section 1.2. Further, some pipeline steps that need to be implemented for improving the efficiency of the data processing have also been described.

1.1 Instrument Improvements

A major feature that can be incorporated into the IEROM to image additional information from the tissue is fluorescence. This will be a major leap for the instrument in terms of the information it can produce. To adapt the IEROM to allow fluorescence imaging of tissue, we need to change to optics train to provide excitation illumination through the objective. The current IEROM is compatible to introduce fluorescence imaging since it has additional room in the optics chain for fluorescence imaging.

Further, the instrument can be improved by incorporating more monitoring and remote response systems. Currently, remote authorization usage and control are limited only to the stage and slicing. However, there is a need to control the water flow and monitor the compressor conditions as well. These features will help the true completely remote operation of an IEROM after its initial setup.

1.2 Data Processing Pipeline Improvements

An important part of the image processing pipeline for any biomedical data set is the registration to standard atlases. Allen mouse brain atlas [18] and Paxinos atlas [58] are some of the standard atlases. “Image registration is the process of overlaying images (two or more) of the same scene taken at different times, from different viewpoints, and/or by different sensors. The registration geometrically align two images (the reference and sensed images)[76].“ We need image registration because it is important to associate the vascular geometry of the brain that has been derived form IEROM with functional areas such as the motor and sensory areas, to give the information meaning. Moreover, only by mapping the imaged data-set to standard data-sets, one can incorporate additional information that is available from other resolutions into the meta-data of the digitized tissue. This will enable a more comprehensive and informative data-set to be available rather than just geometry with annotations that could be hard to discern the various functional parts from.

Other functions that can be added are automation of slice organization from tissue scanner data-sets for the IEROM. Then, automation of the removal of noisy images needs to be done to reduce wastage of processing power down the pipeline and to interpolate meaningful data in place of noise due to improper images. Most importantly, web visualization needs to be added to the pipeline and the analysis engine developed in this paper should be used to analyze and get information that can be annotated into the final online 3D web visualization application. Finally, more functions can be added to analyze the data-sets from the mouse brain vasculature.

1.3 Conclusion

In conclusion, during the course of this thesis an instrument for imaging and scanning brain tissue was setup for operation. Simultaneously, the data pipeline was setup with the necessary steps to create clean data, extract tubular features from the input data into a novel tubular morphology container and to analyze the features extracted. A mouse brain vasculature dataset from BTS was used to test the pipeline and the results are shown here. For better visualization and to enable storing mesh information, a converter was also designed that can take a morphology file input in the SWC format and save it as a VTK Poly-data file.

One method introduced and implemented to visualize teravoxel volumes without loss of accuracy was the divide conquer and combine approach to stitch sub-volumes of the teravoxel data after the tracing is completed. The idea is to have all the relevant data preserved while combining the traced data and removal of redundant data. The hypothesis is that the quality and information that the visualization of this combined trace will provide will be much higher than compressed data that is traced for whole brain feature visualization. The combine algorithm was tested for preliminary results on an 8-volume subset and has the potential to be combined with a re-sampling algorithm to achieve the final visualization goal. Due to requirement of further validation, this proposal has been included as appendix A in this thesis.

To summarize, the assembly and setup of a new instrument called IEROM based on KESM for imaging tissue samples using light microscopy was completed. Further, a preliminary data processing pipeline was setup for preprocessing, feature extraction and analysis of images from the IEROM. The analytical data was mined for a whole mouse brain vasculature dataset and summary of the results were presented in this thesis.

Part V

Appendices

Appendix A

Vasculature Post-Processing

The feature extracted data set in this thesis consists of a lot of individually traced sub-blocks that together make up the volume data. Divide, Conquer and Combine (DCC) algorithm to trace complex tubular structures from teravoxel-sized microscope image stacks was proposed to stitch together these blocks to form the trace of entire dataset. This proposal was accepted as an abstract in ISBI 2016 and is presented here in the current state of work. There are some points of consideration about the practical implementation and use of this idea that will be discussed as further.

A.1 Need for Post-Processing

A large number of light microscopy techniques are available today to gather high resolution data of vasculature and neurites at sub-micrometer resolution [55]. One of the examples is Knife-Edge Scanning Microscopy [44]. This technique uses simultaneous physical sectioning and bright-field line-scan imaging to obtain high resolution images of tissue samples at high throughput rates. Multiple terabytes of data in the form of 2D image stacks is generated from the sectioning and imaging process. Based on the stain used, this data can be either vasculature, neurites or cell bodies.

Chapter 2 describes all the existing methods for vasculature extraction. However, these methods are largely demonstrated only on small unit volumes of 512×512×512pixels and not on teravoxel sized data. So there exists a gap between the need for reconstruction and extraction of tubular morphology data from large teravoxel sized image volumes and the existence of only small-volumes tested tracing algorithms. To solve this, a method was proposed in which the teravoxel volume data is split into sub blocks before tracing, as was done in the feature extraction part of the proposed pipeline. The results of this tracing was separate volumes that each contained parts of the vasculature. However, this might not be ideal for medical researchers who want to view data as a whole.

Most methods that try to address this challenge are either based on manual reconstruction of single structures [23, 75] or on semi-automated methods that rely on experts for manual correction of traces [50, 11, 42]. The first approach is constrained to individual structure tracing while the second approach requires extensive manual work. Both approaches have a fundamental limitation of time, one due to individual analysis in a complete volume and the other due to the extensive manual work required. Fully automated reconstruction thus remains a challenge and it is what is targeted in the method proposed in this paper. Our method proposes to trace complete volumes automatically. One way to do this is to stitch together the data from all the traced volumes by identifying the blood vessels that are connected across the boundaries of every volume on all 6 faces. If it is a corner volume, this would only be 3 faces. The algorithm that was developed for stitching blood vessels at the boundary is explained further over the next sections and the results are illustrated.

A.2 Overview of Proposed Method : Teravoxel Trace Generator

In our method, the data be traced first in smaller volumes using existing tubular structure tracing methods. Then the traced blocks are stitched together using the a novel stitching algorithm. This will enable fast tracing of teravoxel sized data sets.

However, it is important to note that our method is different from *i*mage stitching of multiple sub-volumes prior to tracing [75]. The latter stitches the images itself together. However, in this work the traced data is stitched together rather than the image or volume data This is a subtle but important difference and the former is a distinctively different approach.

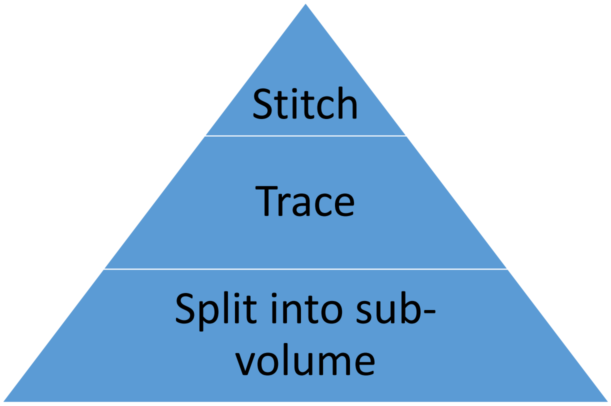


Figure A.1: The method order of execution (Bottom to top).

The overview of proposed method is shown in Figure A.1. The proposed method works by *divide, conquer and combine*. In the *divide* step, adjacent non-overlapping volumes are constructed from the initial teravoxel sized microscopy data. The *conquer* step is to trace the available sub-volumes individually using any technique suited to the structure under consideration. Then, in the *combine* step the traces from adjacent sub-volumes are stitched together. The method assumes that the traced data is available in SWC format [9]. In this work, the Augmented Rayburst Algorithm to trace tubular morphologies was chosen for extracting the trace from the KESM data-set of mouse brain vasculature.

Challenges in Stitching Traced Volumes

There are many challenges in stitching together traced data from adjacent volumes. Some of the most important ones are highlighted below.

- Densely packed structures: The density of tubular structures in light microscopy image stacks is overwhelming, more so in case of blood vessels. The radius of the structures is variable as well. So the algorithm used should be able to choose the right nodes in the sub volume to connect together, usually out of multiple options.

- Trace termination depths: The tracing algorithm used can stop tracing the vessels around the edge at different points in adjacent volumes. This is more so in step-by-step tracing methods and the step size and starting location together will make it different every for the same vessel in adjacent volumes.
- Tracing errors at edges: Many tracing algorithms are sub-optimal and inaccurate near the edges of the volume. So the stitching algorithm has to account for these inaccuracies.

A.3 Implementation of the Trace Stitch Algorithm

Our algorithm iteratively combines traced data in order to get the final stitched trace result. Each SWC file generated by tracing a sub-volume is considered as a *vessel block*. Each entry in an SWC file is a *v*essel node, with a node number, spatial co-ordinates, radius and a parent node number. Each vessel block is a structure containing the array of vessel nodes in the corresponding sub-volume. Apart from this, each vessel block contains meta-information about its relative location in the volume i.e. its ID derived from naming conventions (Section A.3) and also terminal nodes for the block that are derived (Section A.3).

Prerequisite Information

The proposed algorithm requires two important details about the volumes to be available in some format before stitching. These are explained below.

- **B**lock Size: This information tells us the x, y and z dimensions of the original sub-volume from which the given SWC file was generated by tracing.

- **V**olume Location: This is the relative location of the given sub volume in the bigger volume that is required in terms of x, y and z position starting from top left of the final volume.

In the rest of the paper the block size is assumed to be included as a comment at the beginning of every SWC file. It is also assumed that volume ID information is available from the naming convention. Inclusion of this information can be automated using a wrapper code for the tracing algorithm.

Terminal Nodes

To find the terminal nodes at each surface, an initial *threshold* is assumed to be set by the user. In validation tests performed, this parameter was set to probe 20 pixels depth from the surface. Then, the algorithm probes for terminals with a single connection on either side, indicating it to be a terminal node. Depending on the closest surface, the terminal node is added to one or more of the six sets of terminal nodes, one for each face. These vessel nodes carry information about their spatial location and relative location to other nodes in the vessel block they belong to.

Generating Node Pairs

While stitching two sub-volumes, one is considered to be a fixed reference and the other as the volume to append. Surfaces of these sub-volumes in the stitching direction are analyzed to match nodes. Every terminal node on the surface of the fixed sub-volume (say n) is mirrored onto the corresponding surface of the second sub-volume (say m). Then, with the mirrored node as center, a sphere of pre-determined radius is drawn. This radius should be based on the nature of the data set and/or the tracing algorithm used. In validation and testing, radius of 15 pixels was used to cover the farthest node in z direction. All the terminal nodes on the volume m within this sphere are listed as possible matches for the node on volume n. To select the correct node from this list, the algorithm finds the node in m having the least *disparity measure* D with respect to node in n. This measure is defined as given in (A.1) and (A.2).

$$D_{ij} = r_{ij}^2 + d_{ij}^3, \tag{A.1}$$

where d_{ij} is the Euclidean distance from a node in volume n to another node in volume m and r_{ij} is the absolute difference between the radii of the nodes.

$$j = \text{argmin}(D_{ij}), \tag{A.2}$$

where j is a matched node number that has the least D.

Combining Node Pairs

Algorithm 4 Overview of Proposed Algorithm

```
function COMBINE((node1,node2) from (vol1,vol2))
    if node1 is parent && node2 is parent then
        Insert 2 nodes at same point between nodes 1&2
        RadiusOfChildNodes=Avg(node1 & node2 radii)
        node1 = Parent(child1) & node2 = Parent(child2)
    else if node1 is child && node2 is child then
        Insert a node in between the two child nodes
        Inserted node = Parent(node1&node2)
    else                                        ▷ (One node = parent & other has no parent)
        Connect child node to parent node
    return stitched_volume
init : varray = Read traced data into array of structures
stitch_dir ← x,y or z depending on first two blocks
iterations ← $\mathbf{l}og_2$(array.size())
for i in range(iterations) do
    for Every 2 vessel blocks in varray do
        node pairs = Get node pairs from terminals
        combined_volume ← COMBINE(node_pairs)
        Store combined_volume as new SWC file
    varray ← array of SWC files from this iteration
```

Using node pairs obtained by the method in Section A.3, the two vessel blocks can be combined. The nodes in each pair can be of any type and can cause issues without reordering of the node numbers. However, to reorder node numbers is compute intensive, especially as SWC files become larger in size after a few iterations. The different cases are explained here. To stitch two nodes together in an SWC file and also maintain the continuity of the structure, one of the nodes should be a parent with no child and the other should be a child with no parent. In this ideal case, the two nodes can be stitched together by connecting the parent to the child node. However, the cases where both the terminal nodes are parents or both are children has to be handled differently. In the former case, two additional child nodes are created in between the pair in 3D space as earlier. Both these child nodes are at the same physical location but with different node numbers and different parents. In the latter case, a parent node is created that lies in between the two nodes in 3D space and use it to bridge the two volumes. This will help in visualization of the data as a continuous vessel or neurite segment. To ensure that the artificially inserted nodes do not impede and add false information while making quantitative measurements from the traced data, they are labeled with a custom node type in the SWC file. This helps to avoid misinterpreting inserted connector parent nodes as branches or the child nodes as backtracked trace errors. This way, the new SWC files can be saved in a format that enables easy parsing and also continuous stitched visualization.

A.4 Results & Validation

The proposed algorithm was tested and validated on a PC running AMD 64-bit processor at 3.8GHz with 8GB RAM. C++ was used to implement the functionality using Qt Creator IDE. All visualization is done using Vaa3D [60, 59]. The tracing results shown are all done using the MOST tracing plugin [50] in Vaa3D.

For validation, three vessel traces from adjacent sub-volumes from a part of mouse brain vasculature data-set were extracted and modified. This was done using markers in the Vaa3D tool. Then the radii were varied to test on two scenarios that would be common during such traces. Figure A.3 shows a thicker blood vessel and thinner blood vessel crossing very close to each other. The stitch result shows that the proposed algorithm could identify the correct node pairs in this scenario. Another vessel shown validates that the stitching algorithm not only works for angled blood vessels but also for the ones that are approximately parallel to any axes with more displacement between nodes. It is important to note that sensitivity of the algorithm depends on threshold and radius settings.

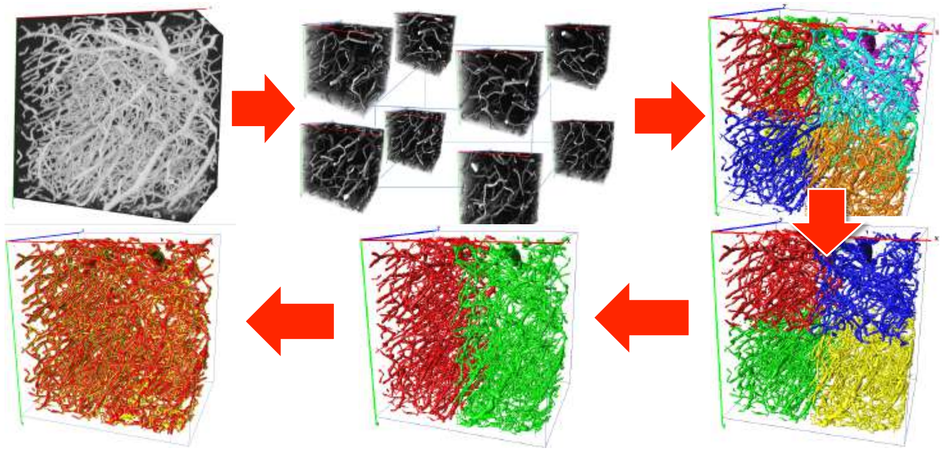


Figure A.2: Flow of proposed algorithm from data to result. Top Left - Original Image Volume. Top Middle - Split the original volume into sub-volumes - 8 in our experiment. Top Right - 8 traced sub blocks (put together in order for visualization). Bottom Right - First iteration combining traced blocks along z-axis to form 4 sub-volumes. Bottom Middle - Second iteration combines traced blocks along y-axis. Bottom Left - Final iteration result(in yellow) overlaid on the direct trace from original volume (in red) with 3-voxel shift in each direction to show the similarity.

Figure A.4 shows a scenario where tracing algorithms could be erroneous near the surface but the proposed algorithm can stitch the traces accurately. This assumes that the tracing method used between two blocks is the same.

The overall flow of implementation in this algorithm and its results on a sub-volume in the KESM whole mouse brain vasculature data-set are shown in Figure A.2. The original volume of data under consideration was divided into 8 subsections. Each subsection was then traced individually before running proposed stitching algorithm on the traced data.

The proposed algorithm iteratively combines the traced blocks by automatically determining stitching direction from volume IDs. The result shows that our algorithm could stitch together the traced results to form the trace of bigger volumes in the data-set automatically with just two required inputs.

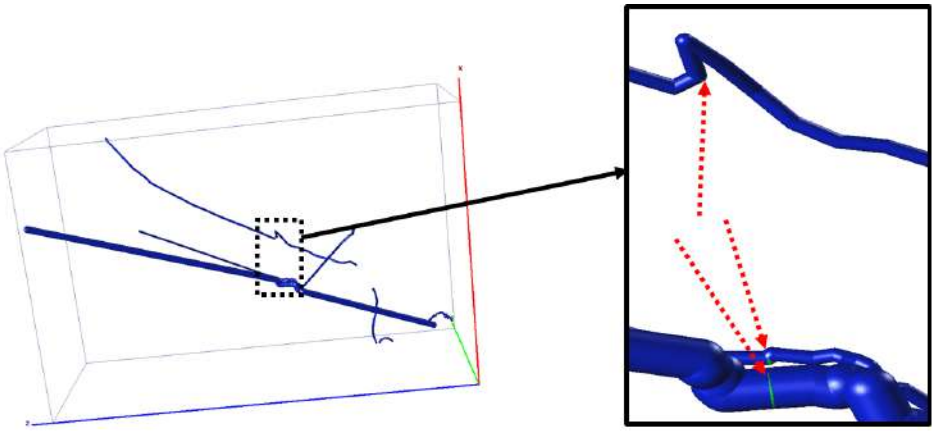


Figure A.3: Testing multiple scenarios. The dotted arrows in the figure to the right show the places where stitching was done by the algorithm. Right Top - Case of displacement between terminal nodes along stitching plane. Right Bottom - Two vessels very close to each other but of different radii are stitched correctly.

A.5 Conclusion & Future Work

A new methodology is introduced for teravoxel tracing of tubular morphologies from KESM data. A novel algorithm is proposed to combine traced sub-volumes from microscopy data of vasculature or neurites to achieve faster trace of teravoxel sized volumes. This will tackle the issue of microscopy data that have larger memory requirements for tracing. Also, if the proposed algorithm is implemented in a parallel processing framework, it will further reduce processing time significantly. We have described in detail the implementation of our stitching algorithm along with pseudo-code and we have shown visualization of our results. Our algorithm will help speed up processing and feature extraction from untraced data of multiple terabytes from light microscopy & similar modalities.

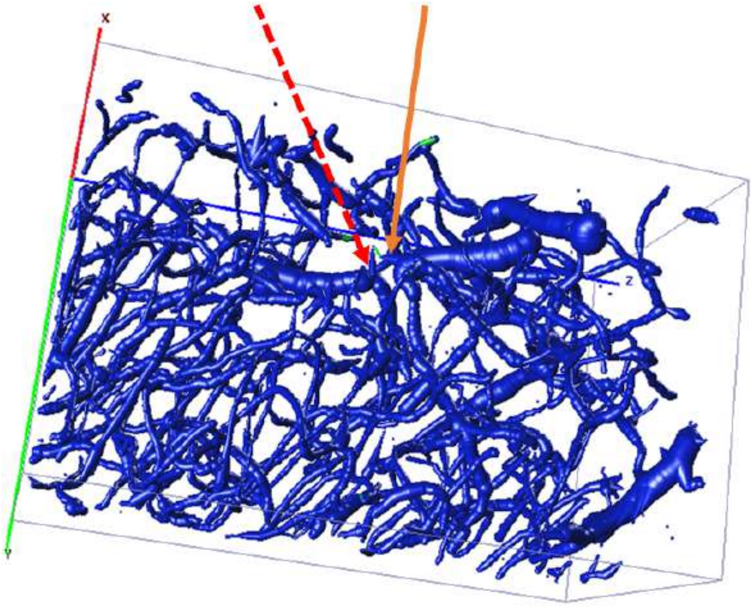


Figure A.4: Special Case - handling inaccuracy of tracing algorithm at edges. The dotted arrow shows terminal of a thick blood vessel on the left sub-volume and solid arrow shows corresponding terminal on the sub-volume to the right. Although the trace is deviant at the surface, the correct vessel traces are connected to each other by the proposed algorithm.

The methodology and algorithm are only in their infancy. Although this research is out there, there are multiple improvements that could be added on to it. For instance, the volume numbering system can be standardized and the algorithm for processing adapted accordingly. Odd number of volumes per dimension should be supported for stitching. There is also the issue of large file size for data as the trace grows larger. To deal with this, re-sampling the SWC file can be done. With the help of the data container proposed as part of the CAETS in Chapter 3, the re-sampling function has been made much easier but it has not been integrated into the Teravoxel Trace Generator yet.

Bibliography

[1] Michael D Abràmoff, Paulo J Magalhães, and Sunanda J Ram. "Image processing with ImageJ". In: *Biophotonics international* 11.7 (2004), pp. 36–42.

[2] James Ahrens et al. *36-ParaView: An End-User Tool for Large-Data Visualization.* 2005.

[3] Giorgio A. Ascoli et al. "Generation, description and storage of dendritic morphology data". In: *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 356.1412 (2001), pp. 1131–1145. ISSN: 0962-8436. DOI: `10.1098/rstb.2001.0905`. eprint: `http://rstb.royalsocietypublishing.org/content/356/1412/1131.full.pdf`. URL: `http://rstb.royalsocietypublishing.org/content/356/1412/1131`.

[4] Automated Imaging Association et al. *Camera Link: Specifications of the Camera Link interface standard for digital cameras and frame grabbers.* Automated Imaging Association, 2007.

[5] Dale L Bailey et al. *Positron emission tomography.* Springer, 2005.

[6] JM Bower and D Beeman. "The book of Genesis: exploring realistic neural models with the GEneral NEural SImulation System. 1998". In: *Telos, Springer, New York* ().

[7] Tim Bray. "The javascript object notation (json) data interchange format". In: (2014).

[8] Mel Brenner, Thomas J. Fellers Stanley Schwartz, and Michael W. Davidson. *Water Immersion Objectives*. 2015. URL: `http://www.microscopyu.com/articles/optics/waterimmersionobjectives.html`.

[9] R.C Cannon et al. "An on-line archive of reconstructed hippocampal neurons". In: *Journal of Neuroscience Methods* 84.12 (1998), pp. 49 –54. ISSN: 0165-0270. DOI: `http://dx.doi.org/10.1016/S0165-0270(98)00091-0`. URL: `http://www.sciencedirect.com/science/article/pii/S0165027098000910`.

[10] John Canny. "A computational approach to edge detection". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 6 (1986), pp. 679–698.

[11] Dmitri B Chklovskii, Shiv Vitaladevuni, and Louis K Scheffer. "Semi-automated reconstruction of neural circuits using electron microscopy". In: *Current opinion in neurobiology* 20.5 (2010), pp. 667–675.

[12] C Chodorow. "Introduction to mongodb". In: *Free and Open Source Software Developers European Meeting (FOSDEM)*. 2010.

[13] Yoonsuck Choe et al. "Charting out the octopus connectome at submicron resolution using the knife-edge scanning microscope". In: *BMC Neuroscience* 11.Suppl 1 (2010), P136.

[14] Yoonsuck Choe et al. "Specimen preparation, imaging, and analysis protocols for knife-edge scanning microscopy". In: *Journal of visualized experiments: JoVE* 58 (2011).

[15] Paarth Chothani, Vivek Mehta, and Armen Stepanyants. "Automated tracing of neurites from light microscopy stacks of images". In: *Neuroinformatics* 9.2-3 (2011), pp. 263–278.

[16] Ji Ryang Chung et al. "Multiscale exploration of mouse brain microstructures using the knife-edge scanning microscope brain atlas". In: *Frontiers in neuroinformatics* 5 (2011).

[17] Richard SC Cobbold. *Foundations of biomedical ultrasound.* Oxford University Press on Demand, 2007.

[18] Hong Wei Dong. *The Allen reference atlas: A digital color brain atlas of the C57Bl/6J male mouse.* John Wiley & Sons Inc, 2008.

[19] Richard O Duda, Peter E Hart, et al. "Pattern classification and scene analysis". In: *J. Wiley and Sons* (1973).

[20] Robert R Edelman and Steven Warach. "Magnetic resonance imaging". In: *New England Journal of Medicine* 328.10 (1993), pp. 708–716.

[21] G. Bard Ermentrout and David H. Terman. "Mathematical Foundations of Neuroscience". In: New York, NY: Springer New York, 2010. Chap. Dendrites, pp. 29–48. ISBN: 978-0-387-87708-2. DOI: `10.1007/978-0-387-87708-2_2`. URL: `http://dx.doi.org/10.1007/978-0-387-87708-2_2`.

[22] Alejandro F Frangi et al. "Multiscale vessel enhancement filtering". In: *Medical Image Computing and Computer-Assisted InterventationMICCAI98.* Springer, 1998, pp. 130–137.

[23] Hui Gong et al. "Continuously tracing brain-wide long-distance axonal projections in mice at a one-micron voxel resolution". In: *Neuroimage* 74 (2013), pp. 87–98.

[24] Matthew Goodman, Ted Blackman, and Ilica Mahajan. *XACT transaction library.* `https://github.com/3Scan/xact-py`. 2016.

[25] Dong Hyeop Han. *Rapid three-dimensional tracing of the mouse brain neurovasculature with local maximum intensity projection and moving windows.* Texas A & M University, 2009.

[26] Donghyeop Han, John Keyser, and Yoonsuck Choe. "A local maximum intensity projection tracing of vasculature in Knife-Edge Scanning Microscope volume data". In: *IEEE International Symposium on Biomedical Imaging.* IEEE. 2009, pp. 1259–1262.

[27] Wenyun He et al. "Automated three-dimensional tracing of neurons in confocal and brightfield images". In: *Microscopy and microanalysis* 9.04 (2003), pp. 296–310.

[28] Moritz Helmstaedter and Partha P Mitra. "Computational methods and challenges for large-scale circuit mapping". In: *Current opinion in neurobiology* 22.1 (2012), pp. 162–169.

[29] Marcela Hernández-Hoyos et al. "Computer-assisted Analysis of Three-dimensional MR Angiograms 1". In: *RadioGraphics* 22.2 (2002), pp. 421–436.

[30] Melonie Heron. "Deaths: leading causes for 2008." In: *National Vital Statistics Reports: From the Centers for Disease Control and Prevention, National Center for Health Statistics, National Vital Statistics System* 60.6 (2012), pp. 1–94.

[31] Melonie Heron. "Deaths: leading causes for 2010." In: *National vital statistics reports: from the Centers for Disease Control and Prevention, National Center for Health Statistics, National Vital Statistics System* 62.6 (2013), pp. 1–96.

[32] Michael L Hines and Nicholas T Carnevale. "The NEURON simulation environment". In: *Neural computation* 9.6 (1997), pp. 1179–1209.

[33] Pieter Hintjens. *ZeroMQ: Messaging for Many Applications.* " O'Reilly Media, Inc.", 2013.

[34] Donna L Hoyert, Jiaquan Xu, et al. "Deaths: preliminary data for 2011". In: *Natl Vital Stat Rep* 61.6 (2012), pp. 1–51.

[35] Jiang Hsieh. "Computed tomography: principles, design, artifacts, and recent advances". In: SPIE Bellingham, WA. 2009.

[36] Cemil Kirbas and Francis KH Quek. "Vessel extraction techniques and algorithms: a survey". In: *Bioinformatics and Bioengineering, 2003. Proceedings. Third IEEE Symposium on.* IEEE. 2003, pp. 238–245.

[37] David Kleinfeld et al. "Large-scale automated histology in the pursuit of connectomes". In: *The Journal of Neuroscience* 31.45 (2011), pp. 16125–16138.

[38] Jaerock Kwon, David Mayerich, and Yoonsuck Choe. “Automated cropping and artifact removal for knife-edge scanning microscopy”. In: *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*. IEEE. 2011, pp. 1366–1369.

[39] Frederic Lauwers et al. “Morphometry of the human cerebral cortex microcirculation: general characteristics and space-related profiles”. In: *Neuroimage* 39.3 (2008), pp. 936–948.

[40] David Lesage et al. “A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes”. In: *Medical image analysis* 13.6 (2009), pp. 819–845.

[41] Chun Hung Li and CK Lee. “Minimum cross entropy thresholding”. In: *Pattern Recognition* 26.4 (1993), pp. 617–625.

[42] Mark H Longair, Dean A Baker, and J Douglas Armstrong. “Simple Neurite Tracer: open source software for reconstruction, visualization and analysis of neuronal processes”. In: *Bioinformatics* 27.17 (2011), pp. 2453–2454.

[43] Raman Maini and Himanshu Aggarwal. “Study and comparison of various image edge detection techniques”. In: *International journal of image processing (IJIP)* 3.1 (2009), pp. 1–11.

[44] David Mayerich, L Abbott, and B McCormick. “Knife-edge scanning microscopy for imaging and reconstruction of three-dimensional anatomical structures of the mouse brain”. In: *Journal of microscopy* 231.1 (2008), pp. 134–143.

[45] BH McCormick, DM Mayerich, and M Wiercigroch. “Nanomachining for high-resolution scanning of mammalian brain microstructure”. In: *Proc. 11th Int. Conf. Frac.* 2005.

[46] Bruce H McCormick. “Design of a brain tissue scanner”. In: *Neurocomputing* 26 (1999), pp. 1025–1032.

[47] Bruce H McCormick. “Development of the brain tissue scanner”. In: *Brain Networks Lab Technical Report* (2002).

[48] Bruce H McCormick. *System and method for imaging an object.* US Patent 6,744,572. 2004.

[49] Bruce H McCormick and DM Mayerich. “Three-dimensional imaging using knife-edge scanning microscopy”. In: *Microscopy and Microanalysis* 10.S02 (2004), pp. 1466–1467.

[50] Xing Ming et al. “Rapid reconstruction of 3D neuronal morphology from light microscopy images with augmented rayburst sampling”. In: (2013).

[51] Thomas B. Moeslund. “Introduction to Video and Image Processing: Building Real Systems and Applications”. In: London: Springer London, 2012. Chap. Image Acquisition, pp. 7–24. ISBN: 978-1-4471-2503-7. DOI: `10.1007/978-1-4471-2503-7_2`. URL: `http://dx.doi.org/10.1007/978-1-4471-2503-7_2`.

[52] Douglas B Murphy. *Fundamentals of light microscopy and electronic imaging.* John Wiley & Sons, 2002.

[53] Francisco PM Oliveira and João Manuel RS Tavares. “Medical image registration: a review”. In: *Computer methods in biomechanics and biomedical engineering* 17.2 (2014), pp. 73–93.

[54] Pavel Osten and Troy W Margrie. “Mapping brain circuitry with a light microscope”. In: *Nature methods* 10.6 (2013), pp. 515–523.

[55] Pavel Osten and Troy W Margrie. “Mapping brain circuitry with a light microscope”. In: *Nature methods* 10.6 (2013), pp. 515–523.

[56] Glenn W Parsons and James Rafferty. “Tag Image File Format (TIFF)-image/tiff MIME Sub-type Registration”. In: (2002).

[57] Erika Pastrana. “Focus on mapping the brain”. In: *Nature methods* 10.6 (2013), pp. 481–481.

[58] George Paxinos and Keith BJ Franklin. *The mouse brain in stereotaxic coordinates.* Gulf Professional Publishing, 2004.

[59] Hanchuan Peng et al. “Extensible visualization and analysis for multidimensional images using Vaa3D”. In: *Nature protocols* 9.1 (2014), pp. 193–208.

[60] Hanchuan Peng et al. “V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets”. In: *Nature biotechnology* 28.4 (2010), pp. 348–353.

[61] Judith MS Prewitt. “Object enhancement and extraction”. In: *Picture processing and Psychopictorics* 10.1 (1970), pp. 15–19.

[62] Shruthi Raghavan and Jaerock Kwon. “Divide, Conquer, and Combine Algorithm to Trace Complex Tubular Structures from Teravoxel-Sized Microscope Image Stacks”. In: *Proceedings of 2016 IEEE International Symposium on Biomedical Imaging: From Nano to Macro.* IEEE. Apr. 2016. DOI: 10.13140/RG.2.1.3539.9440.

[63] Shruthi Raghavan and Jaerock Kwon. “Fully Automated Image Preprocessing for Feature Extraction from Knife-edge Scanning Microscopy Image Stacks - Towards a Fully Automated Image Processing Pipeline for Light Microscopic Images”. In: *Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies.* 2016, pp. 93–100. ISBN: 978-989-758-170-0. DOI: 10.5220/0005708700930100.

[64] Shruthi Raghavan et al. “Internet enabled robotic microscope powered by knife-edge scanning microscopy”. In: *Neuroscience Meeting Planner, New Orleans, LA: Society for Neuroscience.* 2015.

[65] Josh Robinson, Aaron Gray, and David Titarenco. “Getting Started with Meteor”. In: *Introducing Meteor.* Springer, 2015, pp. 27–41.

[66] Johannes Schindelin et al. “Fiji: an open-source platform for biological-image analysis”. In: *Nature methods* 9.7 (2012), pp. 676–682.

[67] Ingrid Scholl et al. “Challenges of medical image processing”. In: *Computer science-Research and development* 26.1-2 (2011), pp. 5–13.

[68] Will Schroeder and Ken Martin. "Bill Lorensen The Visualization Toolkit An Object-Oriented Approach To 3D Graphics, Kitware". In: *Inc. publishers* (2006).

[69] Jrgen Thomas and Thomas Gemming. *Analytical Transmission Electron Microscopy*. Springer Science + Business Media, 2014. DOI: `10.1007/978-94-017-8601-0`. URL: `http://dx.doi.org/10.1007/978-94-017-8601-0`.

[70] Charles Watson, Matthew Kirkcaldie, and George Paxinos. "Chapter 2 - Central nervous system basicsthe brain and spinal cord". In: *The Brain*. Ed. by Charles Watson, Matthew Kirkcaldie, and George Paxinos. San Diego: Academic Press, 2010, pp. 11 –24. ISBN: 978-0-12-373889-9. DOI: `http://dx.doi.org/10.1016/B978-0-12-373889-9.50002-4`. URL: `http://www.sciencedirect.com/science/article/pii/B9780123738899500024`.

[71] Charles Watson, Matthew Kirkcaldie, and George Paxinos. *The brain: an introduction to functional neuroanatomy*. Academic Press, 2010.

[72] Marian Wiercigroch and Erhan Budak. "Sources of nonlinearities, chatter generation and suppression in metal cutting". In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 359.1781 (2001), pp. 663–693.

[73] Brian A Wilt et al. "Advances in light microscopy for neuroscience". In: *Annual review of neuroscience* 32 (2009), p. 435.

[74] Andreas Zakrzewicz, Timothy W Secomb, and Axel R Pries. "Angioadaptation: keeping the vascular system in shape". In: *Physiology* 17.5 (2002), pp. 197–201.

[75] Zhi Zhou, Staci A Sorensen, and Hanchuan Peng. "Neuron Crawler: An Automatic Tracing Algorithm for Very Large Neuron Images". In: *algorithms* 1 (), p. 9.

[76] Barbara Zitova and Jan Flusser. "Image registration methods: a survey". In: *Image and vision computing* 21.11 (2003), pp. 977–1000.