WEB-BASED STEREO RENDERING FOR VISUALIZATION AND ANNOTATION OF SCIENTIFIC VOLUMETRIC DATA

A Thesis

by

DANIEL CHERN-YEOW ENG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2008

Major Subject: Computer Science

WEB-BASED STEREO RENDERING FOR VISUALIZATION AND ANNOTATION OF SCIENTIFIC VOLUMETRIC DATA

A Thesis

by

DANIEL CHERN-YEOW ENG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Yoonsuck Choe
Committee Members,	John Keyser
	Arum Han
Head of Department,	ValerieTaylor

December 2008

Major Subject: Computer Science

ABSTRACT

Web-based Stereo Rendering for Visualization and Annotation of Scientific Volumetric Data. (December 2008)

Daniel Chern-Yeow Eng, B.S., Texas A&M University

Chair of Advisory Committee: Dr. Yoonsuck Choe

Advancement in high-throughput microscopy technology such as the Knife-Edge Scanning Microscopy (KESM) is enabling the production of massive amounts of high-resolution and high-quality volumetric data of biological microstructures. To fully utilize these data, they should be efficiently distributed to the scientific research community through the Internet and should be easily visualized, annotated, and analyzed. Given the volumetric nature of the data, visualizing them in 3D is important. However, since we cannot assume that every end user has high-end hardware, an approach that has minimal hardware and software requirements will be necessary, such as a standard web browser running on a typical personal computer. There are several web applications that facilitate the viewing of large collections of images. Google Maps and Google Maps-like interfaces such as Brainmaps.org allow users to pan and zoom 2D images efficiently. However, they do not yet support the rendering of volumetric data in their standard web interface.

The goal of this thesis is to develop a light-weight volumetric image viewer using existing web technologies such as HTML, CSS and JavaScript while exploiting the properties of stereo vision to facilitate the viewing and annotations of volumetric data. The choice of stereogram over other techniques was made since it allows the usage of raw image stacks produced by the 3D microscope without any extra computation on the data at all. Operations to generate stereo images using 2D image stacks include distance attenuation and binocular disparity. By using HTML and JavaScript that

are computationally cheap, we can accomplish both tasks dynamically in a standard web browser, by overlaying the images with intervening semi-opaque layers.

The annotation framework has also been implemented and tested. In order for annotation to work in this environment, it should also be in the form of stereogram and should aid the merging of stereo pairs. The current technique allows users to place a mark (dot) on one image stack, and its projected position onto the other image stack is calculated dynamically on the client side. Other extra metadata such as textual descriptions can be entered by the user as well. To cope with the occlusion problem caused by changes in the z direction, the structure traced by the user will be displayed on the side, together with the data stacks. Using the same stereo-gram creation techniques, the traces made by the user is dynamically generated and shown as stereogram.

We expect the approach presented in this thesis to be applicable to a broader scientific domain, including geology and meteorology.

To my savior Jesus, my parents, and my sister

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Yoonsuck Choe, for the opportunity to work with him on this idea and for providing guidance and encouragements through out my graduate studies. I would like to thank my committee members, Dr. John Keyser and Dr. Arum Han for their time and their insightful comments on my thesis. I also would like to thank Dr. House and Dr. Srinivasan of Visualization Laboratory at Texas A&M University for partially funding my graduate studies and introducing me to the research area of physical-based modeling and computer graphics. I would like to thank NSF, NIH, NINDS, Texas A&M University Department of Computer Science's Industrial Affiliates Program, and my parents for funding my graduate studies.

I would like to thank my lab mates from Neural Intelligence Lab (NIL) for discussions on research and life issues. I would also like to thank David Mayerich of Brain Networks Lab for KESM data acquisition and discussions regarding the data sets. I must also thank Gen Kazama for his contribution in implementations towards this thesis. I want to thank Huei-Fang Yang of NIL specifically for mentoring me when I was an undergrad research student as well as a first year graduate student. In addition to that, I would like to thank the reviewers and audience at 2008 Texas A&M Student Research Week and 2008 IEEE/EG Volume and Point-based Graphics symposium for valuable comments and suggestions towards this work.

Finally, I would like to thank my friends and church community, without them this process would have been hard. I would like to thank my parents for their constant support and prayer and have always pushed me for the best. In addition to that, they have shown me that learning is a life long process and that I should never give up. I would like to thank my sister for always being there for me, this thesis would have been impossible without her constant support and encouragement.

NOMENCLATURE

KESM	Knife Edge Scanning Microscope
BNL	Brain Networks Laboratory
VRML	Virtual Reality Markup Langauge
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
DOM	Document Object Model
AJAX	Asynchronous JavaScript and XML

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

In recent years, data visualization and volume rendering over the web have been greatly investigated and improved. Web applications such as Google Maps [7] and BrainMaps.org [8] have allowed users to visualize data (such as geographical maps or scans of biological micro-structures) and share knowledge through annotations and discussions. These advancements are a great feat for the general public and the scientific community since it was not possible to rapidly share ideas when such information was only available in printed maps or atlases.

Advancements in high-throughput microscopy technology such as Knife-Edge Scanning Microscopy (KESM) that is developed at the Brain Networks Lab at Texas A&M University is enabling the production of massive amounts of high-resolution and high-quality volumetric data of biological micro-structures. KESM has been used for serial sectioning and imaging of whole mouse brains. In order to tap into the full potential of such a data set, data and model sharing should be made efficient and effective. For example, the massive amounts of data (typically 2 to 20 TB per mouse brain, depending on the microscope objective used [1]) need to be made accessible to researchers around the globe. These researcher in turn should be able to annotate and analyze, and ship their models back to the main server, integrating those into the original data source.

The journal model is *IEEE Transactions on Automatic Control*.

A. Motivation

Given the volumetric nature of the data, visualizing them in 3D is very important. Since we cannot assume that every end user has high-end hardware, an approach that has minimal hardware and software requirements will be necessary, such as a standard web browser running on a typical personal computer. As mentioned earlier, there are various web applications that facilitate the viewing and annotation of 2D images and 2D maps over the Internet such as Google maps and similar applications, but they have yet to support the rendering of volumetric 3D data in their standard web interface. On the other hand, volume visualization over the web has usually been done either through Virtual Reality Markup Language (VRML) or Java Applets [9], both of which are specialized software that might not be available for every end-user. In addition to that, the typical web-based visualization paradigm, which is either a client-side-intensive thick-client or server-side-intensive thin-client might not be a suitable solution to deal with such huge amounts of image data produced by KESM. Here, a new method of rendering volumetric data over the web with a minimal set of required computing resources on both client and server is proposed. The goal of this thesis is to develop a light weight volumetric image viewer using existing web technologies such as HTML, CSS, and JavaScript utilizing the properties of stereo vision to facilitate the viewing and annotations of volumetric data.

The choice of stereogram over other volume rendering techniques is because it allows the usage of raw image stacks produced by KESM without any extra computation on the data at all on the client side and minimal preprocessing operations on the server side. In addition to that, [2] found that binocular depth cues do affect human's judgement of the visual realism of computer generated images. Operations to generate stereo image pairs using 2D image stacks involves 2 visual cues namely distance

attenuation and binocular disparity. By using simple HTML and JavaScript that are computationally cheap, we can accomplish both tasks dynamically in a standard web browser through overlaying 2D images with interleaving semi-opaque layers.

Navigational controls and annotation capability are also important. The web user interface will allow users to pan in the x, y, and z directions as well as zoom in and zoom out of the data set. Due to the volumetric nature of the data, the user interface allows the user to move in the z direction in stacks of multiple images. The web user interface also allows user to make annotations in the form of traces on the image data set. The structure traced by the user will be presented in the form of stereogram next to the original data set. This is to deal with the problem of occlusion caused by overlaying both the image data sets and the markers. We expect the approach presented in this thesis to be applicable to a broader domain, such as geology and meteorology.

B. Scientific Merits

The method being developed in this thesis is a part of a broader vision of attempting to understand the human cortical network. Many other works has been researched and published by Brain Networks Lab at Texas A&M University in this area ranging from physical sectioning microscopy, cortical network connections to 3D geometry reconstruction of mouse brain. See [10] for more information. The mouse brain is chosen because it is genetically similar to the human brain (about 90% identical) [11] thus making it a good alternative when studying the complexity of the human brain. The main contribution of this work toward this vision is the visualization and sharing of mouse brain's data over the web. By sharing the data over the web and allowing collaborations between researchers, we hope that this will be a valuable tool to assist

and speed up the mapping process of the mouse cortical network.

C. Main Results

For this thesis, a light-weight stereo pseudo-3D image viewer with annotation and navigational capability was implemented. Various steps such as pre-processing of the image stacks and image retrieval will also be discussed in this thesis. The scalability of this method will be tested and presented as well. Finally, discussions of limitations and possible future works will be discussed.

D. Thesis Outline

The rest of the thesis is organized as follows. Chapter II will give an overview of KESM and image acquisition through KESM. In Chapter III, related works ranging from volume visualization to stereo imaging will be discussed. In Chapter IV, methods and implementations of the solution will be presented. Both visualization results and scalability of this method will be presented in Chapter V. Discussions and future works will be in Chapter VI. Chapter VII will summarize this work.

CHAPTER II

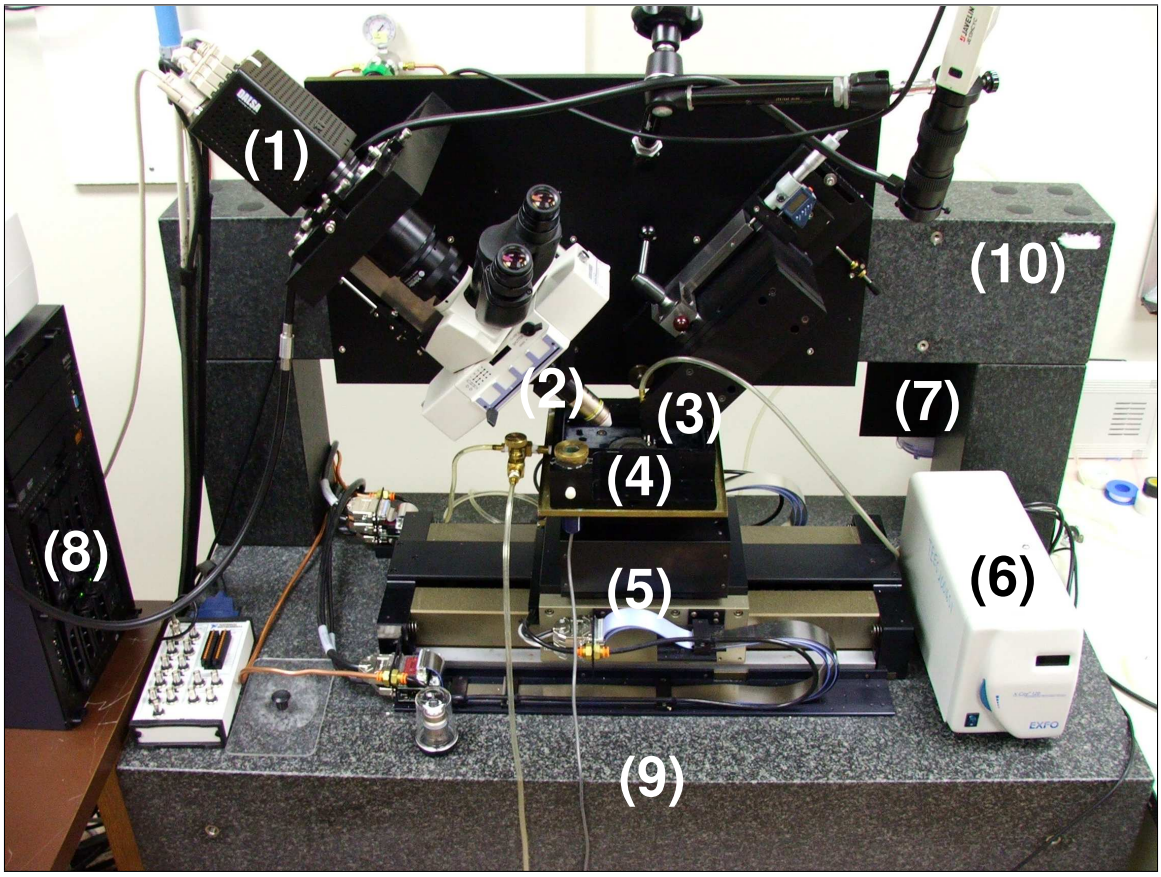
KNIFE-EDGE SCANNING MICROSCOPY (KESM)

A. Introduction

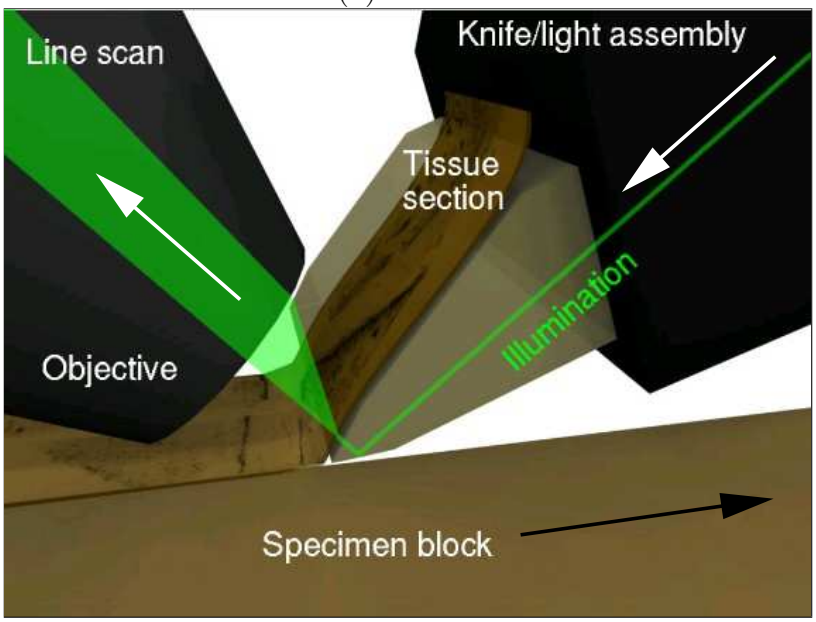
The Knife-Edge Scanning Microscope enables sectioning and imaging of whole small animal organs such as the mouse brain. KESM imaging technique allows the sectioning and imaging operations to be done swiftly, about one sections for every second. The data resolution of the images can be up to 300 nm $\times$ 300 nm $\times$ 500 nm, and when a whole mouse brain is scanned, the resulting data can reach 20 TB. Transmitting such a data set across the network in its entirety is not a viable option, so an on-demand, multi-resolution approach, similar to Google Maps, is necessary.

Figure 1 presents an overall view of the Knife-Edge Scanning Microscope (KESM) and its principle of operation. Figure 1*a* shows a photo of the KESM with its major components marked: (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. Figure 1*b* illustrates the principle of operation of KESM. The objective and the knife are held in place, while the specimen affixed on the positioning stage moves (black arrow) and gets scraped against the diamond knife, generating a thin section flowing over the knife. Line-scan imaging is done at the very tip of the knife where the distortion is minimal. Illumination is provided through the diamond knife (white arrows indicate the light path).

The images obtained from KESM are high contrast and have high quality. Fig-

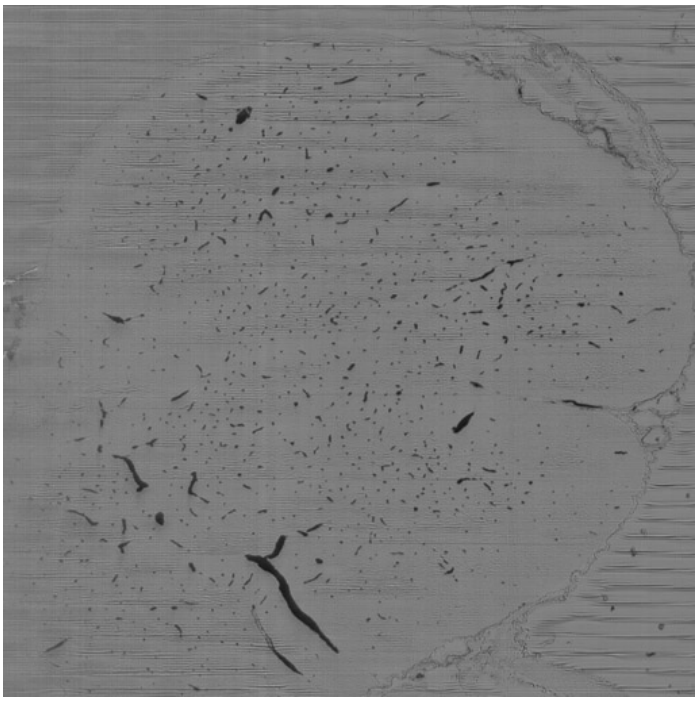


(*a*) KESM



(*b*) Sectioning and imaging

Figure 1: The Knife-Edge Scanning Microscope (KESM). Adapted from [1]. See section A for details.



(*a*) Single raw image slice

(*b*) Volume rendering

Figure 2: KESM data. See section A for details.

Table I: Depth Resolution and Objectives Used.

Data Set	Depth Resolution	Objectives
Spinal Cord	1.0 μm	Nikon 10X
Cortex	1.0 μm	Nikon 10X
Cerebellum	1.0 μm	Nikon 10X

ure 2 shows (*a*) an example image slice and (*b*) a volume rendering of 200 slices (200 μm thick volume, 1 μm thickness per slice) from a mouse spinal cord specimen. The vascular network in the specimen was stained through perfusion using India ink. Black dots of various diameters appearing in figure 2*a* are the blood vessels. A complex web of blood vessels can be seen in figure 2*b*. (Note that the image was rotated and scaled to match the orientation and aspect ratio of *a*.)

B. Data Acquisition

1. Tissue Preparation

The data sets used in this thesis were either stained in India ink or Golgi-Cox solution. Golgi staining reveals the entire structure of neurons, as it stains just 1 % of the neurons in the tissue, individual neurons can be clearly seen. On the other hand, India ink is used to stain the blood vessels and it produces high contrast results. This in turn allows us to remove the background of an image and apply an alpha channel to it, while preserving important information. As mentioned earlier, each image slice has the thickness of 1 μm for all the data sets reported in this thesis. The depth resolution and objectives used for the data set is shown in table 1.

2. Tissue Sectioning

According to [12], a stair-stepping method is used in the sectioning process. This means that every single scanned slide is stored in a squence of square images, most likely 4096 by 4096 pixels. By stacking these images together, image blocks (with 50 - 100 images) can be formed and can be used for 3D reconstruction. Likewise by stacking these images together, we can create a stereo pair as well. Figure 3 illustrates this concept.

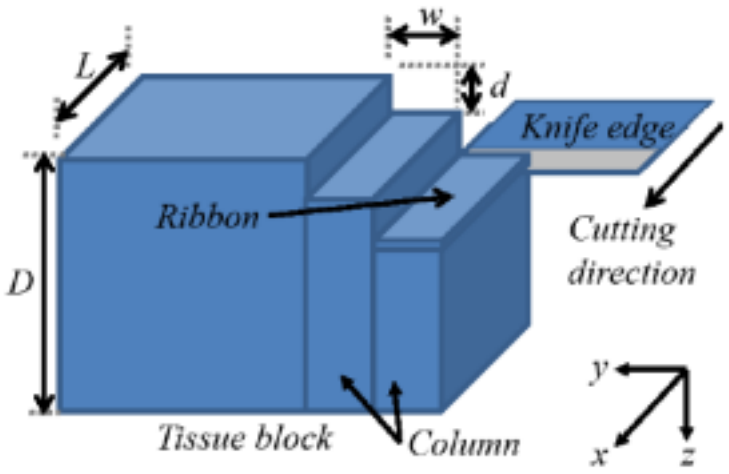


Figure 3: Lateral Sectioning. The knife cuts the tissue block and images the tissue slices across the tissue block.

C. Imaging Results

Examples of results produced by Knife Edge Scanning Microscopy [1] is presented in this section. Figure 4 shows a result produced by Knife Edge Scanning Microscope. The optical resolution along the x and the y axes is 0.3 μm/pixel and the section thickness of 0.5 μm. From the result we can see the fine granularity of the image and the high quality of the image. Also from figure 4 we can learn that a single image

slide does not convey enough information about the structure of the scanned object. However with a stack of multiple images, structure can be revealed.

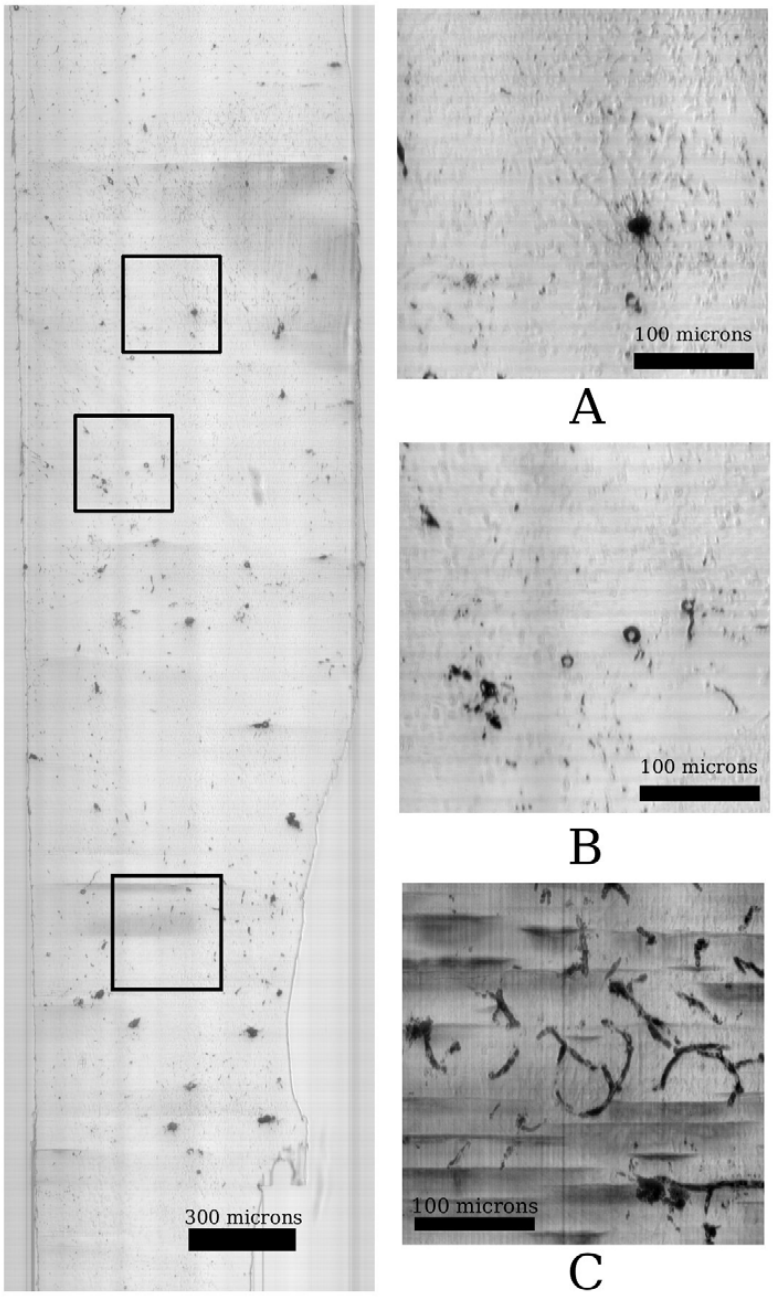


Figure 4: Coronal sections of Golgi-stained mouse cortex. The complete section is shown on the left along with two (A and B) zoomed in version. Image (C) shows an example of combining 20 consecutive sections to show the fiber structures of the cell processes. Adapted from [1].

CHAPTER III

RELATED WORKS

In this chapter, I will give an overview of related works in the area of volume visualization over the web. In addition to that, several aspects of stereo-vision-based rendering such as its realism and rendering techniques will be presented.

A. Volume Visualization Over the Web

In recent years, volume visualization over the web had been a much investigated topic due to various reasons such as advances in detector technology or medical 3D data acquisition devices [13] [14]. It was first presented in the IEEE Visualization 94 conference [9] in the form of client-side visualization, meaning the volume data is downloaded to a local machine via a browser and then processed on the client-side. On the other hand, server-side web-based visualization was made popular through Virtual Reality Markup Language (VRML). As an ISO standard, 3D VRML model has been used to transfer results calculated by the server to the client. In addition to that, Java Applets are usually used to aid the rendering of 3D models on the client side [9]. Increasingly, we have seen creative uses of various optimization techniques that attempted to transmit huge volume of data over the web without costly operations. Web-based remote rendering with Image-Based Rendering Acceleration and Compression (IBRAC) proposed by [15] uses both the server-side and the client-side web-based visualization paradigm. In their algorithm, acceleration through compression is done through an image-based approach, where only images are transmitted through the web and warped for final presentation of the model. The 3D model itself is never transmitted to the client-side thus reducing the latency. The solution is implemented using Java applets. Following that work, [16] implemented the algorithm

using JavaScript, which required little configuration on the client side and require little resources on the client-side. The input for the system proposed by [16] is a set of pre-rendered images. Later, they [17] followed up their work with a multi-resolution approach, where low resolution data are loaded first, and higher resolution data are transmitted when needed, thus increasing the interactivity of their user interface, but they have reverted back to Java Applets in their implementation. Other applications such as [13] also used the mixed model where operations are done on both the client-side and the server-side to increase efficiency. BrainMaps.org [8] on the other hand created a light-weight AJAX viewer that allows users to view (pan and zoom) and annotate image slides of brains produced by standard imaging methods. Their approach is very similar to Google Maps [7]. Although the data sets presented by BrainMaps.org is volumetric in nature, their web interface does not yet support volumetric rendering either in 3D or in pseudo 3D.

From these works, we can definitely see trends in the development. The mixed mode approach of doing most of the computation and compression on the server and then using client-side applications to either render or display the data is fairly common. The mixed mode approaches usually require specialized software such as Java or VRML on the client-side and graphics renderer on the server-side. On the other hand, less computationally intensive approaches require the data set to be confined to two dimensional rendering of the object. An approach that is non-computational intensive on both the server-side and the client-side is much desired when dealing with a huge amount of image data. One possible approach is to tap into the human visual system and offload some operations to it. According to [2], stereo images can be a good alternative when rendering volumetric objects, they stated that "... modern rendering techniques have been criticized as being inefficient in that they compute a lot of information that the human visual system is simply unable to cope with ..."

and "... our results show that it takes at least 20 more seconds to decide on the realism of a scene when using stereo-vision."

Since the images produced by Knife Edge Scanning Microscope have high-resolution and are high-quality, by removing its background (which does not have important information) and stacking them on top of each other will create a minimum intensity projection of the object. The resulting image will look somewhat cluttered and depth ques will be generally missing. But if we overlay the images with intervening semi-opaque layers (see the figure 12 on p.25), distance attenuation can be done and cause image slides closer to the viewer to appear clearer and image slides farther away from the users to appear hazy. By using the same set of images, we can create another stack of images that are slightly shifted thus creating horizontal disparity. Hence a stereo pair can be easily assembled from existing images produced by the Knife Edge Scanning Microscope without intensive calculation on either client or server side. The following sections will give an overview of stereoscopic images and its rendering techniques. Some discussions on the realism of static stereo pairs will be presented as well.

B. Depth Cues

There are several depth cues that we used to determine depth relationship between objects in a scene. According to [18], depth cues can be categorized into two different groups: physiological depth cues and psychological depth cues. Several examples of physiological depth cues include convergence, binocular disparity, and motion parallax. Some examples of psychological depth cues include texture gradient, aerial perspective (distance attenuation), and color. We also know that the effects of depth cue are additive. We can more accurately determine the depth between objects when

more cues are available. The following sections will further explain two depth cues, binocular disparity (which is a physiological depth cue) and distance attenuation (which is a psychological depth cue). With the help of both binocular disparity and distance attenuation, stereo images can be formed, thus allowing data to be perceived in a pseudo 3D fashion.

C. Stereoscopic Images

1. Binocular Disparities

Due to the disparity in the two images perceived by our two eyes caused by their displacement (about 6 cm apart), stereoscopic depth perception is made possible. When such horizontal disparities happen, points in the three dimensional world is slightly different from each of the half images perceived by the eyes. Figure 5 [2] shows a typical setup of our eyes and horizontal disparities. As one fixes both eyes at the object P, image cast by P will fall on both left and right eye, and if object Q in the background is located β degrees from the left eye and α from the right eye, then image Q is said to have disparities of β-α degrees. Thus, the depth (d) is proportional to the distance or degree of separation between P and Q. The resulting images on the left and the right eye shown below in the same figure (figure 5, bottom) gives rise to depth when merged (parallel viewing).

2. Stereo Viewing

According to [3] there are two methods to view a stereo image convergent (or crossed) and divergent (or parallel) viewing. In convergent viewing, users will fix their right eye on the left image, and left eye on the right image while keeping their head straight. The image should converge and will lie in between the two original images. In diver-

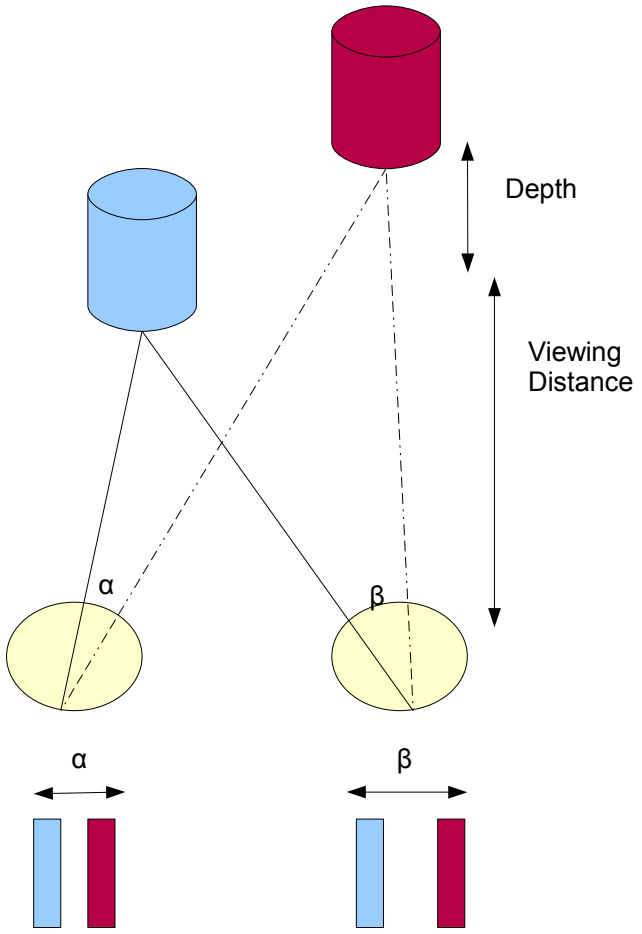


Figure 5: Horizontal Disparity (parallel viewing). Redrawn from [2].

gent viewing, users will fix their right eye on the right image, and their left eye on the left image. Initially the images will look super-imposed and out of focus (when the images are really close to the eyes), as the images move further out, a third image should emerge in between the two images, and it should have convincing depth. There are also several other ways of viewing stereo-gram.

Various other rendering techniques also enhance and aid the perception of depth. One of the common way is through an anaglyph where red channels and blue channels of the stereo pair are imposed on top of each other (see figure 7 in p.19 for an example). A viewing aid with red and blue lenses will then allow the users to view the projected object in 3D. Another popular technique is through stereo animation, where the image perceived by the right eye and the image perceived by the left eye are interchangeably presented to the user in a loop sequence animation. The perception of depth will arise as the images switch back and forth from the right image to the left image. This method is also known as the field sequential techniques [18]. For a more thorough review of commercially available tools such as work bench displays and stereo graphics systems see [18].

3. Realism of Stereoscopic Images

There has been much discussion about how realistic stereo images are and whether they can be a good representation of 3D or volumetric objects. Several studies including [2] and [19] showed that stereo images are perceptually effective for visualizing 3D or volumetric objects. On the other hand, [5] pointed out that the advantages of stereo display exhibited in the above studies might be task-dependent, thus in some tasks stereo viewing might have no perceptual impact at all. Our interpretation is that stereo display is *as good as* other 3D visualization techniques, and thus suitable for our purposes. (Although motion parallax can provide even stronger depth cues,

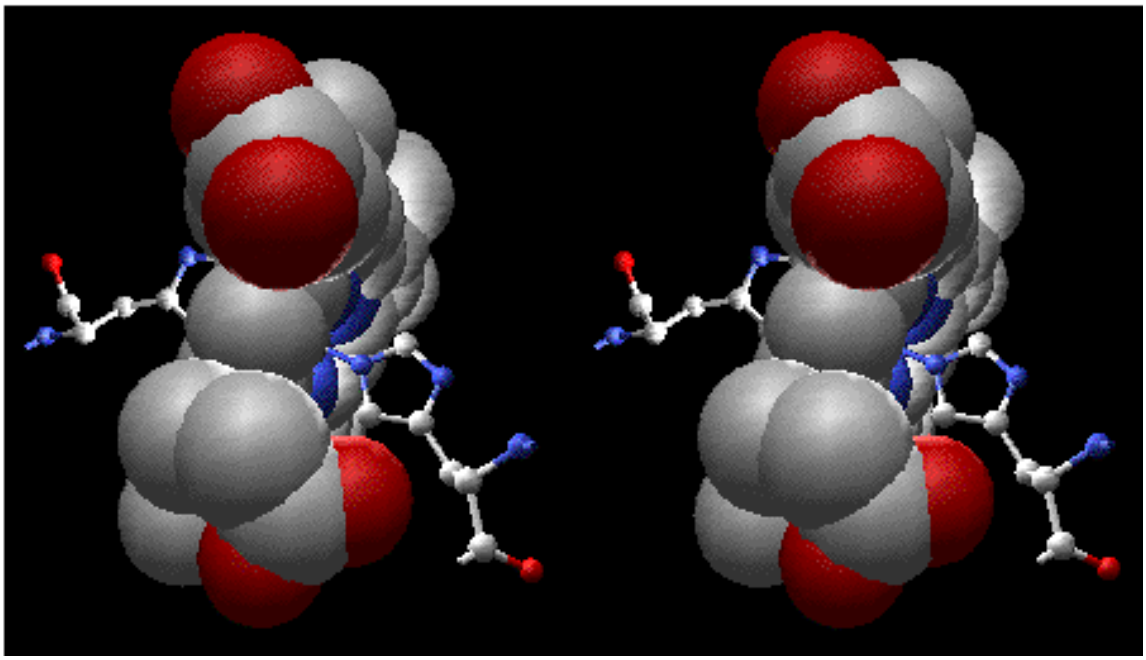


Figure 6: An example of a stereo pair (parallel viewing), from [3].

due to the nature of the static web interface it cannot be used.)

4. Stereo Pairs Rendering Techniques

There are several fast algorithms for generating stereo image pairs (see figure 6 for an example), most prominently [20] and [21]. In both algorithms, they followed the general algorithm of selecting a correct view point for the left-eye image, and then re-projecting it onto the right-eye image. The main difference between the two algorithm is that [20] uses a shear warp algorithm to generate the initial view, while [21] uses a ray casting strategy. The approach that we are taking here is similar to the slice-by-slice rendering method (shear warp volume rendering) mentioned in [20], but since our approach is image based, we are not re-projecting any data to create a new image, we are simply assembling and reassembling pre-existing image slides produced by Knife Edge Scanning Microscope.

Figure 7: An example of an anaglyph, from [4].

D. Distance Attenuation

Distance attenuation can be seen as analogous to aerial perspectives where objects closer to the user looks clearer, while images farther away from the user appears hazy [5]. This effect is also similar to the linear intensity ramp rendering method found in [22]. See the figure 12 on p.25 for an example of distance attenuation and horizontal disparity. See figure 8 for an example of aerial perspectives.

Figure 8: An example of aerial perspectives, from [5].

CHAPTER IV

METHODS

A. General Pipeline

The goal of this thesis is to developed a method to transmit, visualize, and annotate volumetric data over the web using low cost operations by utilizing the properties of human visual system. As far as the pipeline goes, image slides produced by KESM will be reused without any additional 3D modeling or 3D rendering. Inexpensive pre-processing steps are needed to make the images usable for our method. Generally we are just transmitting images over the web and then re-assembling them dynamically as stereo pair image stacks on the client side, thus implicitly using the web browser as our graphics renderer. Figure 9 illustrates this concept.

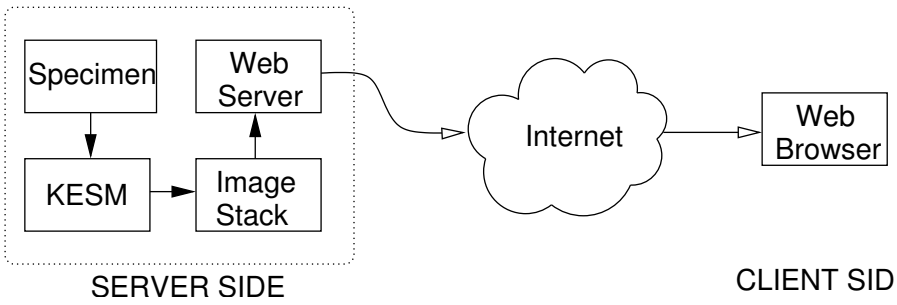


Figure 9: High-level view of the pipeline used in this method. Specimens such as biological micro structures are scanned into stacks of images by the KESM. After some pre-processing, these images are transmitted over the web and is assembled together on the client side by a web browser.

B. Pre-processing

Prior to further processing, the image stack needs to be preprocessed so that overlaying is possible. Alpha channels need to be added to the images for transparency, and background pixels turned transparent. We used command-line tools from ImageMagick (`http://www.imagemagick.org`) for this step. Following is a snap shot of the code that we used to for this preprocessing step.

```
convert $FILENAME -matte -fuzz 10% -transparent
        ["replace with background rgb values"] ${DIR_TEMP}.${EXT}
convert ${DIR_TEMP}.${EXT} +gravity -crop
        ["replace with tile size"] -colorspace Gray
        ${INT}/${DIR_TEMP}_tile_1%04d.${EXT}
```

We first added the alpha channel to the image and then removed the background from the image. Next, we tiled the images into smaller images depending on its initial dimension for faster delivery over the web. Figure 10 illustrates an example of the pre-processing steps.

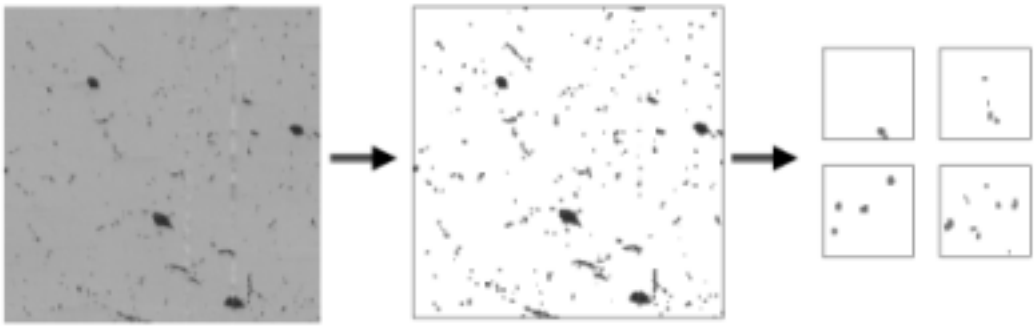


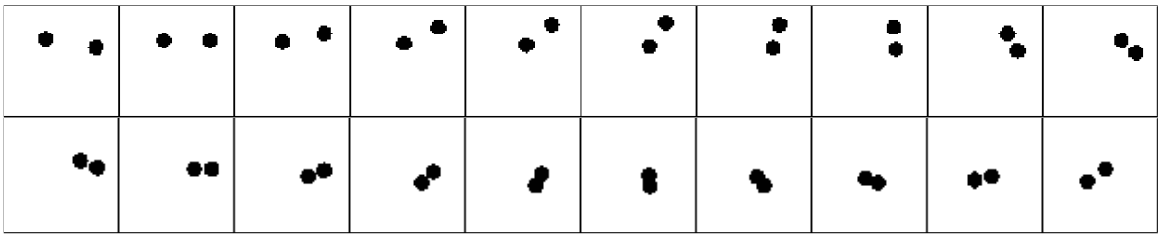
Figure 10: Pre-processing steps done to the image slide produced by KESM.

C. Distance Attenuation

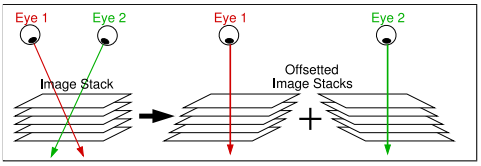
Distance attenuation can be achieved by interleaving white semi-opaque images (typically 5% to 10% opacity dependeing on the size of the stack) in between the data images. This approach has a similar effects as aerial perspectives where objects closer to the image plane appear clearer while objects farther away from the image plane appear hazy. This approach can also be thought of as a variation of the painter's algorithm and image-based rendering where semi-opaque images are added in between images to render depth. One desirable thing here is that we only need a single image for the semi-opaque layer, and reuse that by inserting the same image between the data images (i.e., we only need to download the semi-opaque image once). Figure 11*c*–*d* shows the visual effect achieved by distance attenuation. The depth is much more apparent with distance attenuation (*d*) than without (*c*), even without stereo merging. The resulting image stacks will provide the depth cue that is similar to the linear intensity ramp method and accumulation of buffers method mentioned in [22]. But, since our method is image based, no extra computation is needed to re-render the images. Figure 12 shows the the concept and assembly of an image stack with distance attenuation.

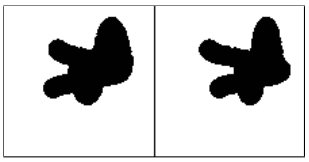
D. Horizontal Disparity

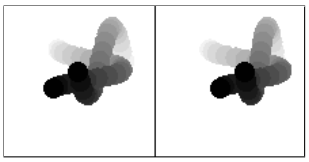
Horizontal disparitiy can be achieved using an overlaying technique similar to distance attenuation. Instead of stacking each image directly on top of each other, the horizontal position of each image can be offsetted slightly. Figure 11 shows a simple example with a synthetic volume image stack. Figure 11*b* shows how shearing the image stack can result in a stereo pair that gives a pseudo 3D effect. When offsetting the images, the depth of focus can be adjusted. For example, in figure 11*d*, the depth



(*a*) Raw (synthetic) data slices







(*b*) Generating 3D views (*c*) Offset image stacks (*d*) Dist.-atten./offs. image stacks

Figure 11: Pseudo Stereo Pair Generation and Distance Attenuation. Three-dimensional effects can be generated using a simple overlaying method. (*a*) A series of 20 image stacks taken from a synthetic volume data set is shown (left to right, top to bottom). (*b*) An illustration of how two stereo pairs can be generated using simple offsetting (shearing) and overlaying of the image stack. Such overlays can be easily produced in a web browser using Cascading Styling Sheets (CSS), from base images with transparencies (alpha channel). (*c*) Stereo-pair (for cross viewing) of offset image stacks are shown. The three dimensional effect is weak due to ambiguities in stereo registration. (*d*) Stereo-pair (for cross viewing) of offset image stacks with distance attenuation is shown. This can be achieved easily by inserting white semi-opaque layers interleaved between the data images. The three-dimensional structure of the embedded object is clearly visible. Adapted from [6]

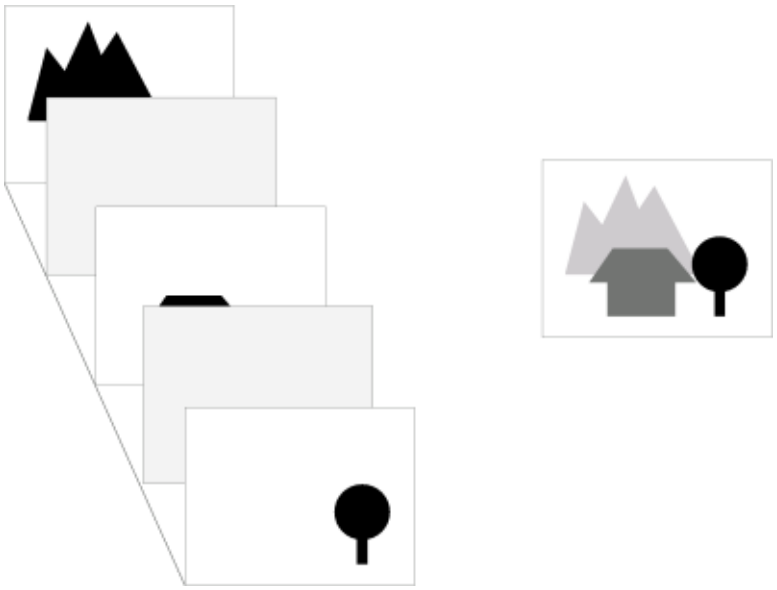


Figure 12: Distance attenuation by interleaving the data set with semi-opaque layers.

of focus is at the far background, thus images at the bottom of the stack will not be shifted while the images on the top of the stack will be shifted the most. Note that the horizontal disparity effect is much more vivid when combined with distance attenuation, as can be seen from figure 11*c* and *d*. It is noted that depth perception arises when the stereo pair is merged.

E. Implementation Techniques

1. Transparency over the Web

Transparency over the web is made possible through images in the format of Portable Network Graphics (PNG) [23] which is now supported by major web browsers such as Internet Explorer 7 and Firefox. PNG is a loss-less image format that allows the full usage of the alpha channel which is significantly better than JPEG images that only supports binary alpha channel. The images produced by KESM are in the TIFF format but they are converted to PNG during the pre-processing steps.

2. HTML Implementation

To create two image stacks in HTML, we simply use the division `<div>` tags to represent two image stacks. Through the manipulation of the Document Object Model (DOM) using JavaScript we will be able to dynamically add images into these divisions. Since the position of these divisions are absolute, we will be able to specify the location of each image, thus allowing us to overlay them or offset them before overlaying an image on top of another image [24]. The following is the listing of the actual code used in the HTML file.

```
<div id="imgcontent" position ="absolute">
    <div id="imgstacks" position ="absolute">
    </div>
    <div id="imgoffsets" position ="absolute">
    </div>
</div>
```

3. JavaScript Implementation

We implemented several operations in JavaScript. The first thing we implemented is the fetching routine where we load the images transmitted from the web server to the browser's cache. Then we implemented the overlaying routine where new images are stacked on top of each other using the absolute position property found in Cascading Styling Sheets (CSS). To generate offsetted image stacks, we simply offsetted the position of the image by N pixels either to the right or to the left depending on the choice of the user (for parallel or crossed viewing). To stack an image on top of another image, we dynamically generated an image element and then set its position to an absolute position using the CSS properties, before appending the element back

to the HTML document. It should be noted that even though we are generating two stacks of images, the images are only loaded once, but used twice, thus reducing the memory required to store the images.

The following is the actual JavaScript code we used to implement this functionality. We first create an image element, populate its attributes through JavaScript and attached it to the HTML DOM tree. Since the position of the image is absolute, we are able to stack multiple images on top of each other.

```
//create a new image tag and update its source
var newimg=document.createElement('img');
newimg.src=currImageArray[counter].src;
// set the style sheet properties of the image tag
// set absolute positioning
newimg.style.position='absolute';
// specify the location of the image and its size
newimg.style.top = toppos;
newimg.style.left = leftpos;
newimg.style.width = width;
newimg.style.height = height;
// append the newly created image tag to the DOM tree
document.getElementById("imgstacks").appendChild(newimg);
//create a new image tag and update its source
var opagueimg = document.createElement('img');
opagueimg.src = OpaqueImage.src;
// set the style sheet properties of the image tag
// set absolute positioning
```

```
opagueimg.style.position='absolute';
// specify the location of the image and its size
opagueimg.style.top = toppos;
opagueimg.style.left = leftpos;
opagueimg.style.width = width;
opagueimg.style.height = height;
// append the newly created image tag to the DOM tree
document.getElementById("imgstacks").appendChild(opagueimg);
```

To offset the position of each image slides, we just update the position of the image element by adding an offset value to its position.

```
//create a new image tag and update its source
var newimg=document.createElement('img');
newimg.src=currImageArray[counter].src;
// set the style sheet properties of the image tag
// set absolute positioning
// specify the location of the image and its size
// the left position of the image slide is slightly shifted by the offset value
newimg.style.position='absolute';
newimg.style.top = toppos;
newimg.style.left = leftpos + offsetValue;
newimg.style.width = width;
newimg.style.height = height;
// append the newly created image tag to the DOM tree
document.getElementById("imgoffsets").appendChild(newimg);
```

F. Web User Interface

The web interface to navigate and annotate the data set is vital for the visualization of the volumetric data. Intuitive navigational controls and annotation methods will increase the usage and usefulness of the data set. In the following sub-section, the design and implementation of the web interface will be discussed. Annotation method and its implementation will be presented as well.

1. Overview

The web user interface is layout in the following fashion, see figure 13 for the layout. At the top of the page, area (1) allows users to choose a data set that they want to view. Area (2) allows users to choose the opacity of the interleaving slides (1, 3, 5, 10, or 20 % opacity). Area (3) is where the data set is displayed. Annotations can be made on the left image stack in area (3) and will be projected to the correct location on the right image stack. Area (4) is where the annotation/trace structures are shown without occlusion caused by the images from the data set. Area (5) allows users to choose a marker for their annotation. Area (6) displays information and metadata about the data set (if available). Area (7) contains the navigational controls. Area (8) displays annotation information.

2. Navigation

Due to the volumetric nature of the data, the ability to navigate in all x, y, and z directions is very important. This means that users should be allowed to move in the z direction in a stack of multiple images limited to a certain range, thus reducing the need to load the entire set all at once, which can take up a lot of memory. In addition to that, due to the granularity of the object, as well as the level of details

needed by the users to view or share data, zooming capability becomes important as well. Last but not least, users will also have the ability to update the disparity of the right image stacks to better suit their stereo viewing needs.

3. Annotation Implementation

Annotation is vital to the development of this project. Again, due to the volumetric nature of the data, and the specific geometry of the data, a conventional scheme of annotation on maps such as placing a push pin icon will not work very well. We believe that a user must has the capability to make traces on the data set and then annotate those traces. Since the data set in presented in the form of image stacks, some trace markers might be occluded by newly added layers. Using the same method that generates a stereo image pair, the markers of the different traces are displayed next to the data set as another stereo pair. This means that the user can dynamically generate pseudo 3d structures by tracing the structures in the data set while viewing the result as a separate stereo pair on the side. See figure F for an example.

The user will have the ability to export the traces onto an XML file with the following format:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<AnnotationXML>
  <Label>Annotation 1</Label>
  <Desc>Annotation 1's Description </Desc>
  <Point x = "100" y = "100" z = "0"/>
  <Point x = "200" y = "100" z = "1"/>
  <Point x = "102" y = "100" z = "2"/>
  <Point x = "50" y = "100" z = "3"/>
```

```
  <Point x = "104" y = "50" z = "4"/>
</AnnotationXML>
```

Likewise, an XML file with the same format can be loaded into the viewer and the traces will appear in the correct z layer.

Figure 13: Web User Interface. See section F in chapter IV for more information.

CHAPTER V

RESULTS

In this chapter, several results will be presented. First, the visualization results where the results produced by our method is compared to the volume rendering and maximum intensity projection of the object. In addition to that, screen shots of annotations and navigation operations will be presented. Finally, the measurements of the memory usage and download time are provided.

A. Visualization Results

The method outlined in the previous chapter was applied to three data sets produced by KESM. The cortex and cerebellum data sets were stained using the Golgi-cox solution while the spinal cord data set was stained using India-ink. Cortex and cerebellum data sets consisted of about 300 image slides (thus 300 μm in depth), while the spinal cord data set had only 49 images (it was the initial test case). The results alongside with original samples and anaglyph versions are presented in the following sections. In each subsection, a single slide from the data set is shown to re-enforce that 2D image slide alone does not convey enough information about the structure scanned by KESM. In addition to that, volume rendering and maximum intensity projection of the object are presented. Finally, image stacks with distance attenuation, stereo pairs and a corresponding anaglyph version are presented. It is clear that our method is a good alternative to full volume rendering of an object, given limited bandwidth and computing resources.

1. Cortex

Figures 14, 15, 16, 17, 18, and 19 show the results from the Golgi-stained mouse cortex data set. From these figures we can conclude that our method is better when compared to the maximum intensity projection of the volumetric data. In addition to that, our method allows user to zoom in and out of the data set to find specific or general features of the object. However, our approach does not allow dynamic change in the view point.

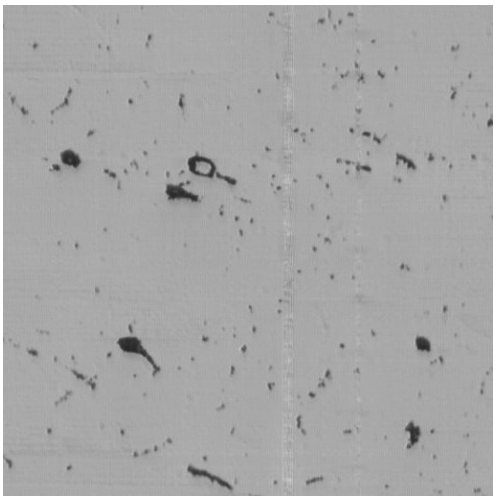


Figure 14: Golgi-stained mouse cortex data set: single image slide.

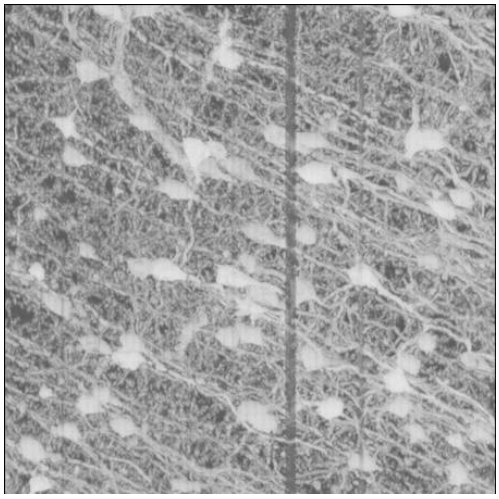


Figure 15: Golgi-stained mouse cortex data set: maximum intensity projection (307 slides).

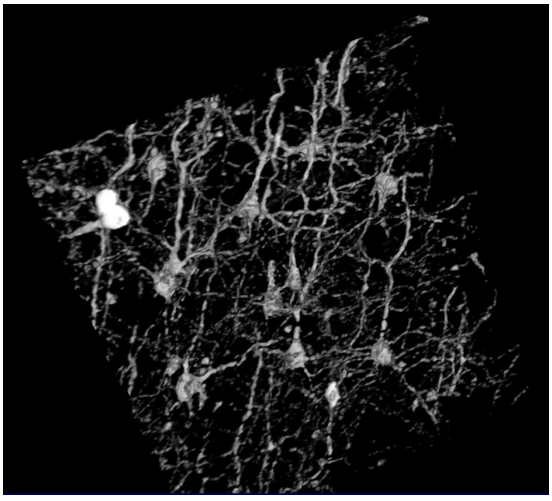


Figure 16: Golgi-stained mouse cortex data set: volume rendering.

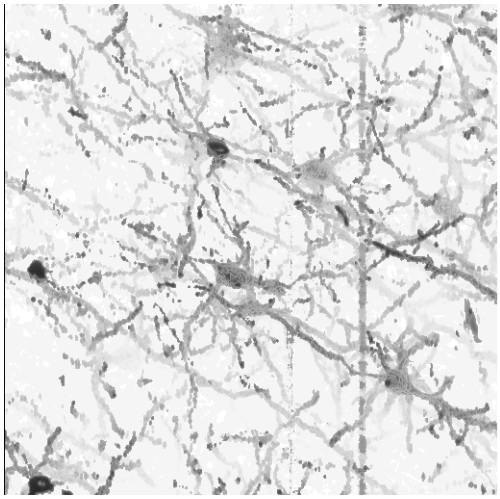


Figure 17: Golgi-stained mouse cortex data set: image stack with distance attenuation (307 slides).

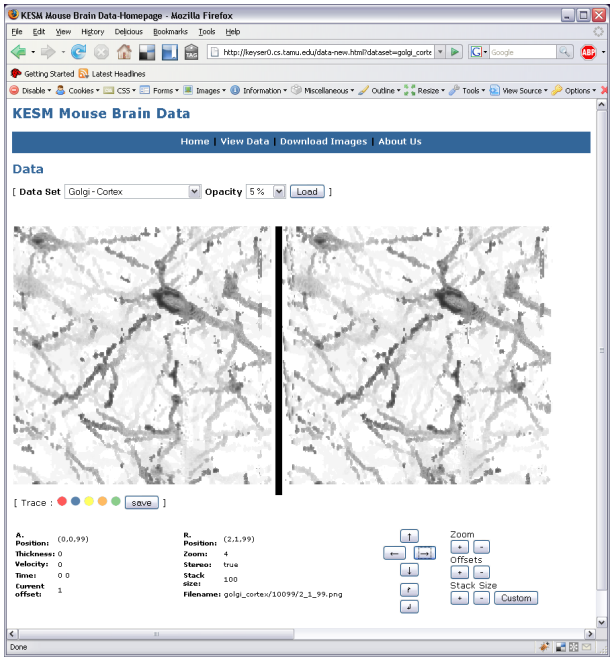


Figure 18: Golgi-stained mouse cortex data set: stereo pairs shown in web browser for crossed viewing (100 slides).

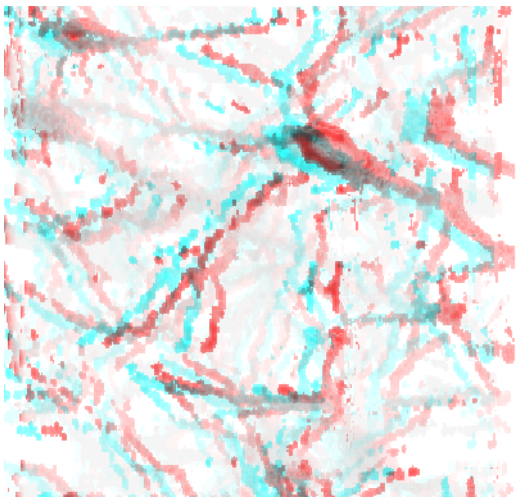


Figure 19: Golgi-stained mouse cortex data set: anaglyph (red-left, blue-right) generated using stereo pair from Figure 18.

2. Cerebellum

Figures 20, 21, 22, 23, 24, and 25 show the results from the Golgi-stained mouse cerebellum data set. For this data set, the volume rendered version gives a better view since it was generated from a good view point where the Purkinje cell's dendrites could be seen in full view. However, our method is better than maximum intensity projection.

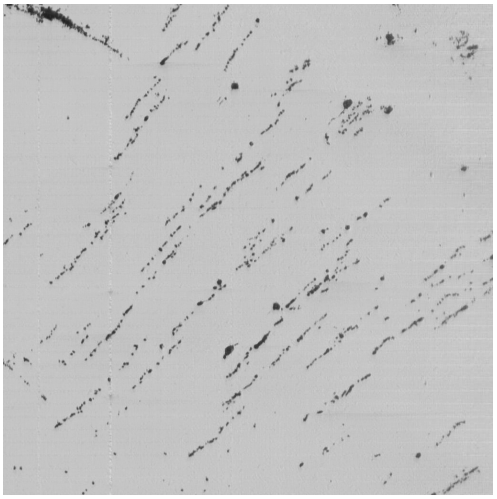


Figure 20: Golgi-stained mouse cerebellum data set: single slide.

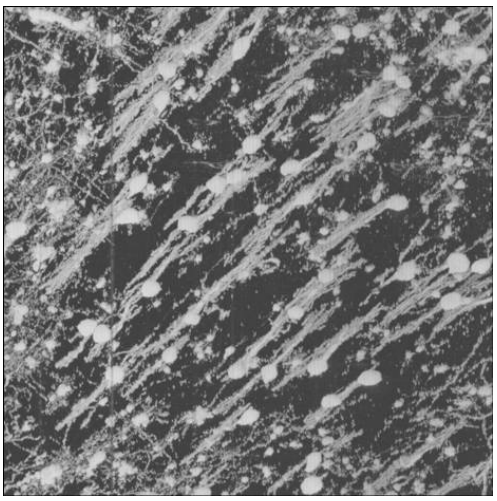


Figure 21: Golgi-stained mouse cerebellum data set: maximum intensity projection (303 slides).

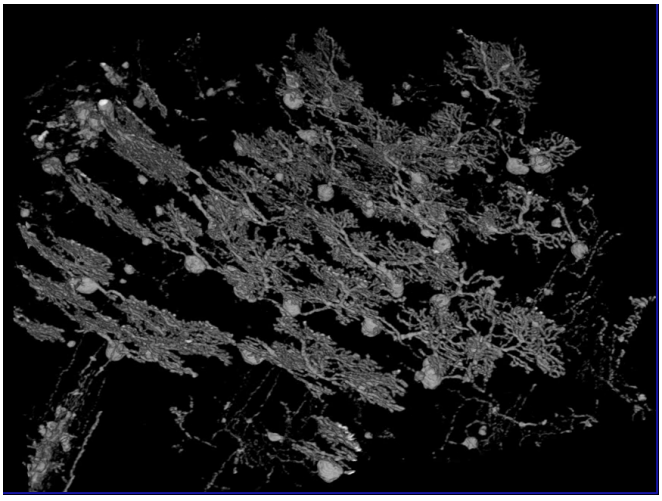


Figure 22: Golgi-stained mouse cerebellum data set: volume rendering.

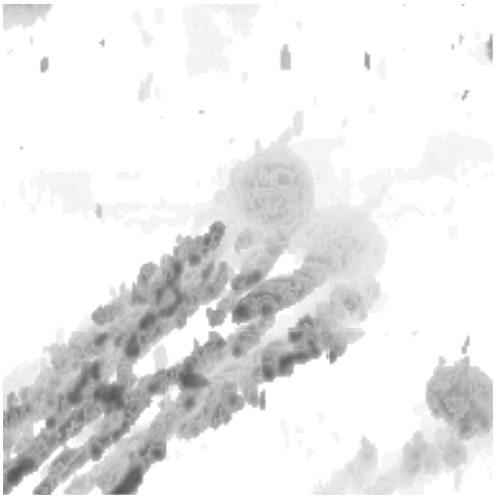


Figure 23: Golgi-stained mouse cerebellum data set: image stack with distance attenuation (50 slides).

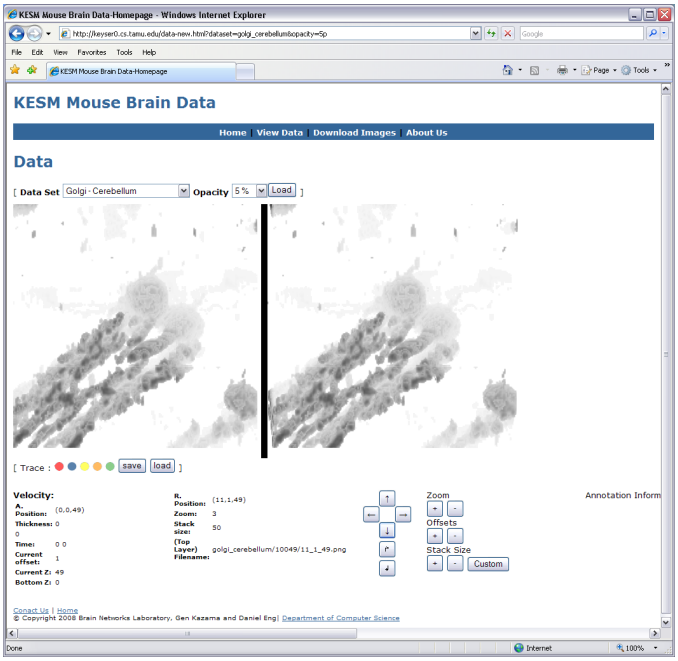


Figure 24: Golgi-stained mouse cerebellum data set: stereo pairs shown in web browser for crossed viewing (50 slides).

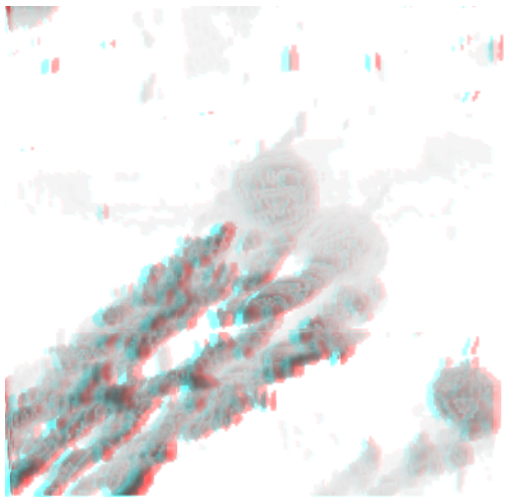


Figure 25: Golgi-stained mouse cerebellum data set: anaglyph (red-left, blue-right) generated using stereo pair from Figure 24.

3. Spinal Cord

Figures 26, 27, 28, 29, 30, and 31 show the results from the India-ink-stained mouse's spinal cord data set. In this data set, through India-ink staining, vascular structure of the spinal cord is clearly revealed. Even though our method only uses 49 slides, its result is comparable to the volume rendered version of the data. Major features of the spinal cord can be seen and traced using our method.

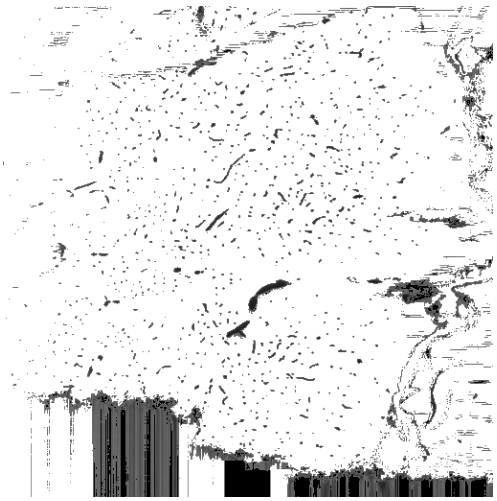


Figure 26: India-ink-stained mouse spinal cord data set: single slide.

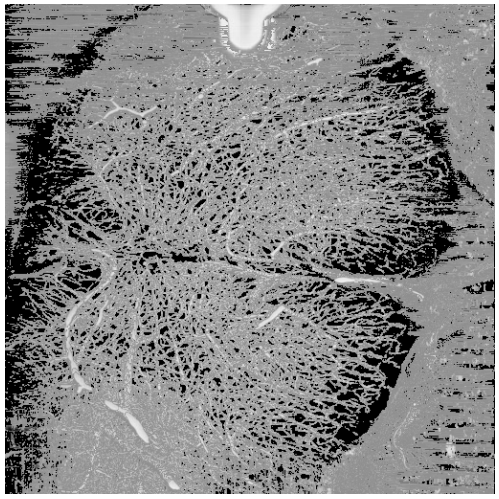


Figure 27: India-ink-stained mouse spinal cord data set: maximum intensity projection (49 slides).

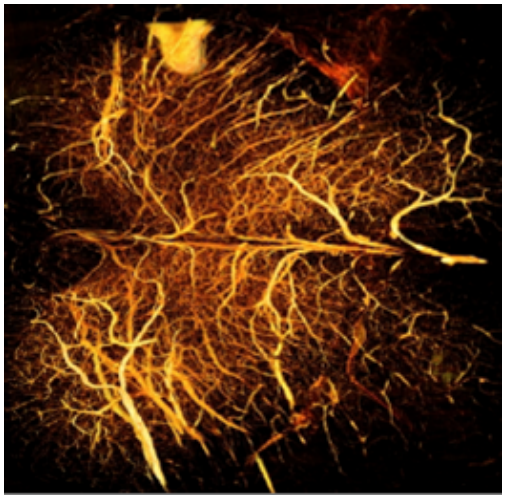


Figure 28: India-ink-stained mouse spinal cord data set: volume rendering.

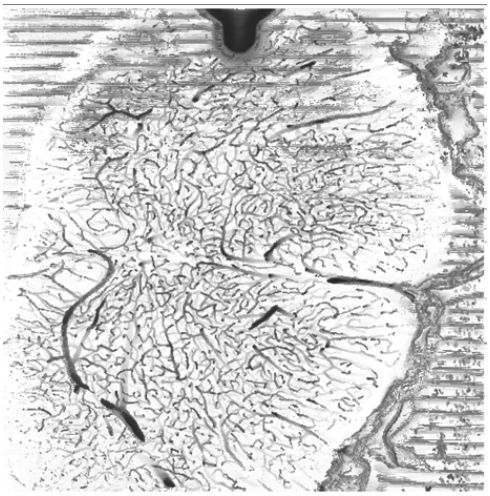


Figure 29: India-ink-stained mouse spinal cord data set: image stack with distance attenuation (49 slides).

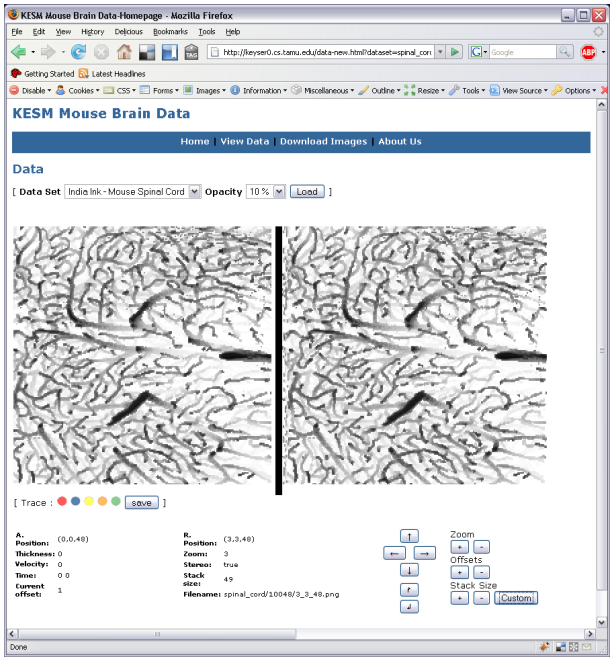


Figure 30: India-ink-stained mouse spinal cord data set: stereo pairs shown in web browser for crossed viewing (49 slides).

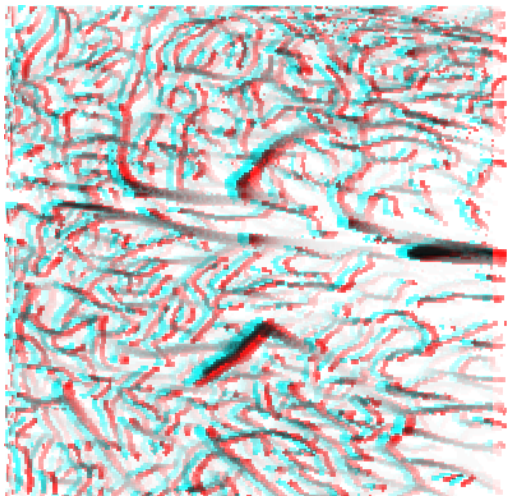


Figure 31: India-ink-stained mouse spinal cord data set: anaglyph (red-left, blue-right) generated using stereo pair from Figure 30.

4. Annotation

Figures 32 and 33 show the example of dynamically generated structures when annotating the data set. Annotations made by the user on the left most image stack (which is not shifted at all) will be dynamically projected to the right image stack (which is shifted). The same marker will also be drawn in the "annotation space" next to the data set to allow users to view the traced structure without occlusion due to the data set.

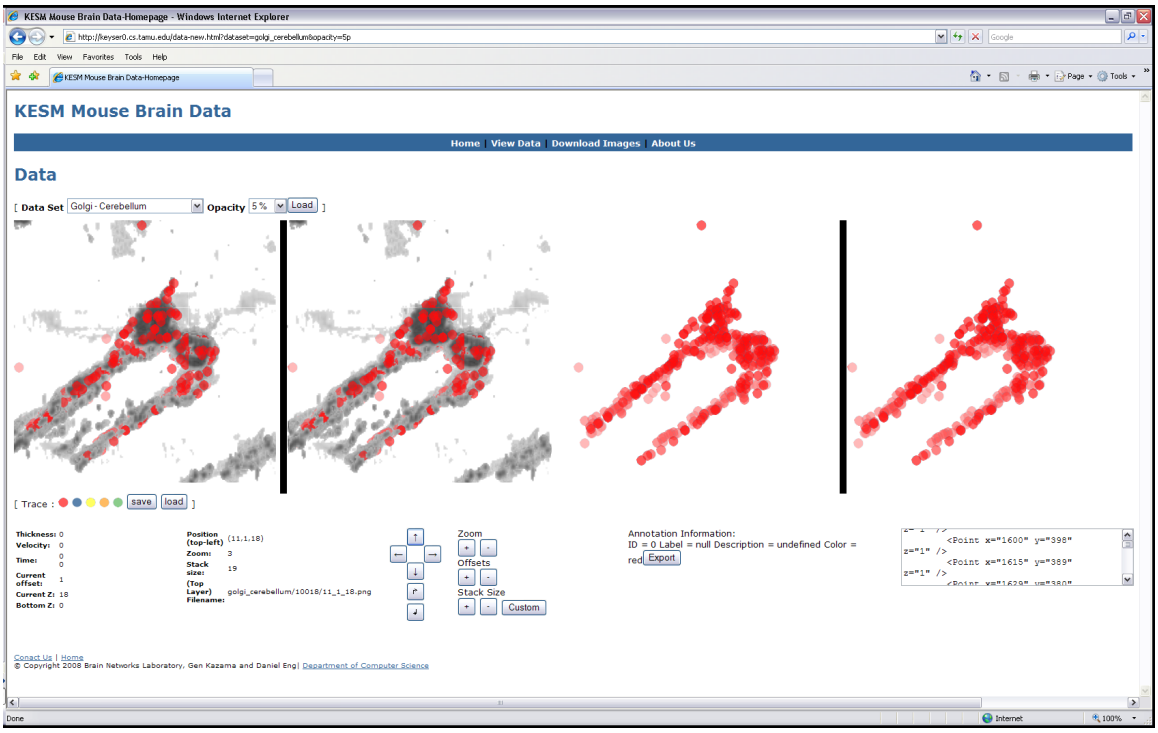


Figure 32: Annotation: stereo pair in web browser.

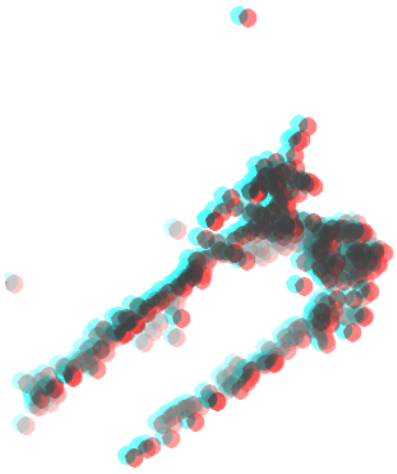


Figure 33: Annotation: gray anaglyph (red-left, blue-right) generated using stereo pair from Figure 32.

B. Scalability Measurements

Several measurements are taken to show that our method is scalable for any size of volumetric data. Both the download time and the memory usage of this method were tested in two popular web browsers, Mozilla Firefox 2.0 and Microsoft Internet Explorer 7.0. For the memory usage, we measured how much memory was needed to load image stacks with 15, 30, 50 and 100 layers. For image load time, we measured how much time is required to load 15,30 and 50 images. A linear growth in image load time and memory usage is expected since no specialized algorithm is used to compress the images. Moreover, with proper garbage collection scheme, we also expect that the memory usage to stay the same when panning through a data set with a constant amount of layers (note : this scheme was evident in Internet Explorer 7.0, but not in Firefox 2.0).

Figures 34 and 35 shows the memory usage and web page loading time. First, we tested the memory usage when displaying image stacks of 15, 30, 50 and 100 layers. Again, Firefox 2.0 (red squares) and Internet Explorer 7.0 (blue diamond) were tested. As expected, the memory usage scales linearly. Next, we measured the download time for 15, 30, and 50 image slices. For this test, we used the *Web Page Speed Report* at `http://www.websiteoptimization.com`. Two types of networks were tested, ISDN 128Kbps (blue diamond) and T1 1.44Mbps (red square), and both show linear scaling property. For T1 connections, downloading even 500 images (10 times the amount reported in our results) could be done in a reasonable amount of time. Once the images are downloaded (and put in the browser cache), displaying the overlayed images takes only several seconds (3 seconds on a 1.7 GHz PC running Linux, with 1GB RAM), so that different display parameter values can be tested.

We also compared the memory usage of other common applications and web sites

to that of our web application. We find that the memory usage of our implementation is comparable to other typical applications. As shown in table B the memory usage of the image viewer loaded in Firefox with 50 loaded layers is actually comparable to the memory usage of common desktop applications such as an email client. In addition to that, table B also shown that the relative memory required to run the image viewer is just slightly higher than the initial load of 2D image map viewers such as Google Maps or BrainMaps.org. Thus, from there we can conclude that, even with a modest amount of images (such as 50) in the stack used to generate a stereo pair, the performance of the browser should not slow down too much, thus making our approach suitable even for lower-end machines. As the garbage collection scheme in JavaScript engine gets better, we might get an even better performance.

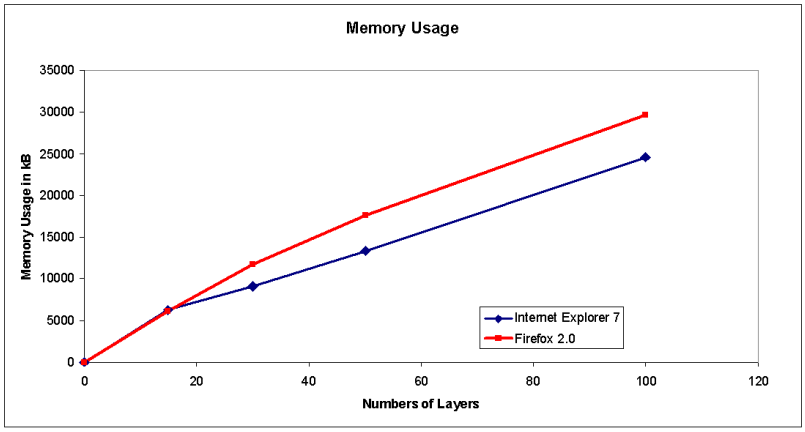


Figure 34: Memory usage.

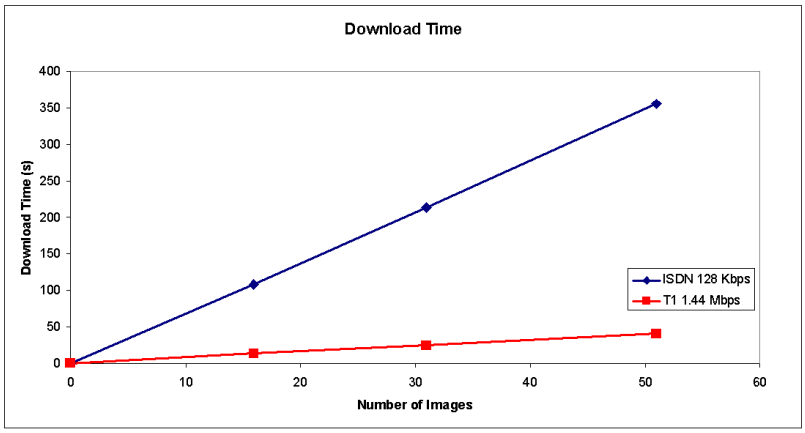


Figure 35: Download time.

Table II: Absolute Memory Usage of Application.

Application	Memory Usage (MB)
Google Talk	54.7
Mozilla Thunderbird (email client)	64.8
Firefox 2 (with 50 layers and our image viewer)	58.3
Internet Explorer 7 (with 50 layers and our image viewer)	162.2

Table III: Relative Memory Usage of Websites.

Web Site	Memory Usage (MB)
TAMU Homepage	6.8
Google Maps (initial load)	9.2
BrianMaps.org (initial load)	11.6
Our Image Viewer (loaded with 50 layers)	17.6

CHAPTER VI

DISCUSSION

By using common web technologies and stereopsis we have enabled the visualization, navigation, and annotation of volumetric data over the web. Through basic HTML, JavaScript, and CSS, pseudo 3D stereo pairs were dynamically assembled directly from 2D image stacks. These operations are computationally cheap and do not require intensive calculation on either the server or the client.

We were able to render complex micro-structures with the only bottleneck being the number of images and their sizes (cf. [16]). Our main contribution is to have shown an effective method for 3D visualization of large volume data through standard web protocols, with minimum hardware and software requirements. Even though most computers today are equipped with modest graphics capabilities, we believe that the images and data generated by 3D microscopes will continue to exceed their capacities, thus our method can be a good alternative for viewing of such data.

Our approach is ideal for delivering large biological volume data sets such as those from the mouse brain. There are notable efforts in building web-based brain atlases [25, 26, 27, 8]. Among these, BrainMaps.org [8] is the closest to our approach. The main difference between our approach and BrainMaps.org is that we allow the viewing of multiple stacked images at once. We allow the user to view the volumetric structures in a pseudo 3D way while BrainMaps.org only allows the user to view one image slide at a time. BrainMaps.org also has a desktop 3D visualization application [28] called StackVis. StackVis allows users to download image slides from their server and stacks them on top of each other. In StackVis, distance attenuation and transparency are not introduced as depth cues, as a result the image stack will look cluttered and confusing. StackVis does not support full 3D volume rendering thus

the shapes of individual brain structures are not explicitly shown. Also, StackVis is a platform dependent application, it is currently available only on Windows and Linux (through Wine, an open source implementation of Windows API). There are other approaches for web-based 3D visualization, such as [13], [29] and [17], but unlike these, our approach requires no specialized software or hardware; just a standard web browser that supports CSS and JavaScript is all that is necessary. Our approach also differs from [16], [17], and [15] because we do not render a 2D image model on the server-side before transmitting it over to the web browser. This means that we do not need to render the model on the server-side every time a request is sent, we are just re-assembling the raw images on the client-side, thus making our approach not operationally intensive on the server-side.

There are several limitations of our current approach. The obvious one is the requirement that users should be able to do stereo merging, and extended viewing (especially in the crossed viewing case) can put a lot of stress on the eyes. Simple and cheap viewing aids made of mirrors or prisms, or similar optical components can be used to overcome this issue. These tools are discussed in both [30] and [18]. Another issue is that the stereo pairs generated by offsetting are only pseudo accurate, not fully accurate and has several limitations [31]. A major issue that cannot be easily addressed is that of interactivity. Unlike 3D visualization using full hardware/software support, our approach cannot generate different viewpoints on the fly: the viewpoint is predetermined by how the image stack is organized and cannot be changed. Also, one might argue that 300 images are not enough to represent a large volume of data. As shown in the results section, our approach scales linearly, so adding more images is not a problem given a sufficient amount of memory. Depending on the number of images, the opacity of the semi-opaque layers can be reduced. Another limitation of our approach is the memory usage of the web browser. Since we are not doing

any explicit compression when delivering images, viewing more than 100 layers of images might slow the web browser down. A possible solution will be generating pre-merged images (5, 10, or more images stacked and merged into a single image) when larger, lower-resolution images are being viewed. This way, deeper volumes can be viewed with fewer images. Another possible solution will be using images with lower resolution in the bottom of the stack thus reducing the memory usage. In addition to that, better garbage collection scheme in the JavaScript engine or explicit implementation of garbage collection functionality can further improve the memory usage by freeing images that are currently not in the viewing window from the browser cache.

There are several future extensions that can be applied to this work. One of them is to allow the usage of personal annotations where users are allowed to upload their own annotation through an XML file and download a copy of their annotations in XML format when they are done with their traces. Other existing technologies such as Google Gears [32] that utilize a local SQL database might be a better solution to deal with both on-line and off-line content. By further implementing our method using Google Gears we might be able to let users load their annotations from their machines while loading images from our server and vice-versa. In addition to that, capabilities such as removing a marker dynamically and adding tool tips to the annotation should be considered. A better scheme of data transmission such as the inclusion of data compression will speed up the transmission process. Algorithms that could reduce the usage of memory can be included. Providing a 2D map of the morphology of the object can also be very helpful. Adding more depth cues to the stereo pair can be helpful. For example, the use of color to distinguish depth should be considered. Having a scale bar on the image itself will also give users extra information when determining the depth of the object. Also, a usability study on how effective and

accurate our method is should be conducted. Results from the usability study can help us further improve the user interface with regard to annotation and navigation. Also, based on the results from the usability study, we might be able to improve the accuracy of our method.

Finally, to work toward the goal of mapping out the connections of the mouse cortical network, annotations submitted by the researchers can be further use for other purposes such as feature identification through pattern recognition. Although it is beyond the scope of this thesis, one might think that the structures traced by various researchers can be applied and learned using pattern recognition algorithms to allow the identification of similar structures that appear elsewhere in the cortical networks. Also, another extension can allow the sharing of model and data through a peer-to-peer network thus allowing models to be shared rapidly and transmitted effectively over the web.

CHAPTER VII

SUMMARY

By exploiting how the human visual system works regarding 3D perception, this work showed that it is possible to develop an efficient visualization method for the distribution, visualization, and annotation of volumetric data over the Internet using a standard web browser. Our method showed that depth and horizontal disparity effects can be dynamically generated over the Internet using a combination of HTML, CSS, JavaScript, and an original image stack data. Scalability measurement also showed that these operations are computationally efficient, with a linear scaling property with regard to the number and size of the image slides. The main differences between the presented method and methods found in the literature are that operations used to pre-process and transmit the data are not operationally intensive and are platform independent while allowing 3D volumetric rendering. We expect our approach to provide an efficient alternative to specialized 3D visualization techniques, and allow dissemination of large volumes of volumetric data to a larger population of researchers.

REFERENCES

[1] D. Mayerich, L. C. Abbott, and B. H. McCormick, "Knife-edge scanning microscopy for imaging and reconstruction of three-dimensional anatomical structures of the mouse brain," *Journal of Microscopy*, vol. 231, 2008.

[2] C. H. Lo and A. Chalmers, "Stereo vision for computer graphics: the effect that stereo vision has on human judgments of visual realism," in *SCCG '03: Proceedings of the 19th Spring Conference on Computer Graphics*, New York, NY, April 2003, pp. 109–117.

[3] G. Rhodes, "Stereo viewing," http://www.usm.maine.edu/r̃hodes/0Help/StereoView.html, September 1997.

[4] United States Department of the Interior, "3d geology tour of saguaro national park," http://3dparks.wr.usgs.gov/saguaro/html/w51.html, April 2003.

[5] M. Kersten, J. Stewart, N. Troje, and R. Ellis, "Enhancing depth perception in translucent volumes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1117–1124, 2006.

[6] Y. Choe, L. C. Abbott, D. Han, P. Huang, J. Keyser, J. Kwon, D. Mayerich, Z. Melek, and B. H. McCormick, "Knife-edge scanning microscopy: High-throughput imaging and analysis of massive volumes of biological microstructures," in *High-Throughput Image Reconstruction and Analysis: Intelligent Microscopy Applications*, A. Ravi Rao and G. Cecchi, Eds. Boston, MA : Artech House, 2008, In press.

[7] Google, "Google maps," http://maps.google.com, October 2007.

[8] S. Mikula, I. Trotts, J. M. Stone, and E. G. Jones, "Internet enabled high-resolution brain mapping and virtual microscopy," *NeuroImage*, vol. 35, no. 1, pp. 9–15, 2007.

[9] K. Brodlie and J. Wood, "Volume graphics and the internet," in *Volume Graphics*, M. Chen, A. Kaufman, and R. Yagel, Eds., pp. 317–331. London: Springer, February 2000.

[10] Brain Networks Lab, "Brain networks lab," http://research.cs.tamu.edu/bnl/, May 2008.

[11] S. Russell, "Of mice and men striking similarities at the DNA level could aid research," in San Francisco Chronicle, December 2002.

[12] D. Mayerich, "Acquisition and reconstruction of brain tissue using knife-edge scanning microscopy," M.S. thesis, Texas A&M University, College Station,TX, December 2003.

[13] S. Prohaska, A. Hutanu, R. Kahler, and H.-C. Hege, "Interactive exploration of large remote micro-ct scans," in *VIS '04: Proceedings of the Conference on Visualization '04, IEEE Computer Society*, Washington, DC, October 2004, pp. 345–352.

[14] M. Agus, F. Bettio, E. Gobbetti, and G. Pintore, "Medical visualization with new generation spatial 3d displays," in *Eurographics Italian Chapter Conference*, Eurographics Association, Trento, Italy, February 2007.

[15] I. Yoon and U. Neumann, "IBRAC: Image-based rendering acceleration and compression," *Eurographics*, vol. 19, pp. 321–330, 2000.

[16] J. Chen, E. W. Bethel, and I. Yoon, "Interactive internet delivery of scientific visualization via structured prerendered imagery," *Internet Imaging VII*, vol. 6061, 2006.

[17] J. Chen, I. Yoon, and W. Bethel, "Interactive, internet delivery of visualization via structured prerendered multiresolution imagery," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 2, pp. 302–312, 2008.

[18] D. F. Mcallister, "Display technology: Stereo & 3d display technologies," in *Wiley Encyclopaedia on Imaging Science and Technology*, J. P. Hornak, Ed., pp. 1327–1344. Bognor Regis, West, United Kingdom: Wiley Interscience, 2005.

[19] W. Bethel and S. J. Bastacky, "Measurement of perceived objects," in *IEEE Visualization 99, Late Breaking Hot Topics*, 1999.

[20] T. He and A. Kaufman, "Fast stereo volume rendering," in *VIS '96: Proceedings of the 7th Conference on Visualization '96, IEEE Computer Society Press*, Los Alamitos, CA, October 1996, pp. 49–ff.

[21] S. J. Adelson and C. D. Hansen, "Fast stereoscopic images with ray-traced volume rendering," in *VVS '94: Proceedings of the 1994 Symposium on Volume Visualization*, New York, NY, October 1994, pp. 3–9.

[22] W. Heidrich, M. McCool, and J. Stevens, "Interactive maximum projection volume rendering," in *VIS '95: Proceedings of the 6th Conference on Visualization '95, IEEE Computer Society*, Washington, DC, 29 Oct-3 Nov 1995, p. 11.

[23] G. Roelofs, "Portable network graphics," http://www.libpng.org/pub/png/, August 2008.

[24] J. Croft, "Creative use of png transparency in web design," http://www.digital-web.com/articles/web_standards_creativity_png/, May 2007.

[25] E. S. Lein, M. J. Hawrylycz, N. Ao, M. Ayres, A. Bensinger, A. Bernard, A. F. Boe, M. S. Boguski, K. S. Brockway, E. J. Byrnes, L. Chen, L. Chen, T.-M. Chen, M. C. Chin, J. Chong, B. E. Crook, A. Czaplinska, C. N. Dang, S. Datta, N. R. Dee, A. L. Desaki, T. Desta, E. Diep, T. A. Dolbeare, M. J. Donelan, H.-W. Dong, J. G. Dougherty, B. J. Duncan, A. J. Ebbert, G. Eichele, L. K. Estin, C. Faber, B. A. Facer, R. Fields, S. R. Fischer, T. P. Fliss, C. Frensley, S. N. Gates, K. J. Glattfelder, K. R. Halverson, M. R. Hart, J. G. Hohmann, M. P. Howell, D. P. Jeung, R. A. Johnson, P. T. Karr, R. Kawal, J. M. Kidney, R. H. Knapik, C. L. Kuan, J. H. Lake, A. R. Laramee, K. D. Larsen, C. Lau, T. A. Lemon, A. J. Liang, Y. Liu, L. T. Luong, J. Michaels, J. J. Morgan, R. J. Morgan, M. T. Mortrud, N. F. Mosqueda, L. L. Ng, R. Ng, G. J. Orta, C. C. Overly, T. H. Pak, S. E. Parry, S. D. Pathak, O. C. Pearson, R. B. Puchalski, Z. L. Riley, H. R. Rockett, S. A. Rowland, J. J. Royall, M. J. Ruiz, N. R. Sarno, K. Schaffnit, N. V. Shapovalova, T. Sivisay, C. R. Slaughterbeck, S. C. Smith, K. A. Smith, B. I. Smith, A. J. Sodt, N. N. Stewart, K.-R. Stumpf, S. M. Sunkin, M. Sutram, A. Tam, C. D. Teemer, C. Thaller, C. L. Thompson, L. R. Varnam, A. Visel, R. M. Whitlock, P. E. Wohnoutka, C. K. Wolkey, V. Y. Wong, M. Wood, M. B. Yaylaoglu, R. C. Young, B. L. Youngstrom, X. F. Yuan, B. Zhang, T. A. Zwingman, and A. R. Jones, "Genome-wide atlas of gene expression in the adult mouse brain," *Nature*, vol. 445, pp. 168–176, 2007.

[26] Allen Institute for Brain Science, "Allen brain atlas," http://www.brain-map.org/, May 2008.

[27] A. MacKenzie-Graham, E. S. Jones, D. W. Shattuck, I. D. Dinov, M. Bota, and

A. W. Toga, "The informatics of a C57BL/6J mouse brain atlas," *Neuroinformatics*, vol. 1, pp. 397–410, 2003.

[28] I. Trotts, S. Mikula, and E. G. Jones, "Interactive visualization of multiresolution image stacks in 3d," *NeuroImage*, vol. 35, no. 3, pp. 1038–1043, April 2007.

[29] K. Engel, R. Westermann, and T. Ertl, "Isosurface extraction techniques for web-based volume visualization," in *VIS '99: Proceedings of the Conference on Visualization '99, IEEE Computer Society Press*, Los Alamitos, CA, October 1999, pp. 139–146.

[30] H. A. Mallot, *Computation Vision : Information Processing in Perception and Visual Behavior*, Cambridge, MA: Bradford Book, MIT Press, 2001, Translated from German by John S. Allen.

[31] Whitman Richards, "Structure from stereo and motion," *J. Opt. Soc. Am. A*, vol. 2, no. 2, pp. 343–349, 1985.

[32] Google, "Google gears," http://code.google.com/apis/gears/, September 2008.

VITA

Daniel Chern-Yeow Eng was born in Batu Pahat, Johor, Malaysia. At the age of fifteen he moved to Houston, TX with his family and graduated from Clear Lake High School in 2002. He received his Bachelor of Science degree in computer engineering from Texas A&M University in December 2006, and received his Master of Science degree in computer science from Texas A&M University in December 2008.

Permanent Address:

Department of Computer Science and Engineering, 3112, TAMU,

College Station, TX 77843-3112, USA.

E-mail Address : edaniel1984@gmail.com

The typist for this thesis was Daniel Chern-Yeow Eng.