

2014

Visualization and Analysis Tools for Neuronal Tissue

Michael Morehead
West Virginia University

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Morehead, Michael, "Visualization and Analysis Tools for Neuronal Tissue" (2014). *Graduate Theses, Dissertations, and Problem Reports*. 294.
<https://researchrepository.wvu.edu/etd/294>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Visualization and Analysis Tools for Neuronal Tissue

by

Michael Morehead

Thesis submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Computer Science

George A. Spirou, Ph.D.
Yenumula V. Ramana Reddy, Ph.D.
Gianfranco Doretto, Ph.D., Chair

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2014

Keywords: Visualization, Neuroscience, Three-Dimensional Graphics, Virtual Reality, User Interfaces, Head Mounted Displays, Stereoscopic Vision

Copyright 2014 Michael Morehead

Abstract

Visualization and Analysis Tools for Neuronal Tissue

by

Michael Morehead

Master of Science in Computer Science

West Virginia University

Gianfranco Doretto, Ph.D., Chair

The complex nature of neuronal cellular and circuit structure poses challenges for understanding tissue organization. New techniques in electron microscopy allow for large datasets to be acquired from serial sections of neuronal tissue. These techniques reveal all cells in an unbiased fashion, so their segmentation produces complex structures that must be inspected and analyzed. Although several software packages provide 3D representations of these structures, they are limited to monoscopic projection, and are tailored to the visualization of generic 3D data. On the other hand, stereoscopic display has been shown to improve the immersive experience, with significant gains in understanding spatial relationships and identifying important features. To leverage those benefits, we have developed a 3D immersive virtual reality data display system that besides presenting data visually allows augmenting and interacting with them in a form that facilitates human analysis.

To achieve a useful system for neuroscientists, we have developed the BrainTrek system, which is a suite of software applications suited for the organization, rendering, visualization, and modification of neuron model scenes. A middle cost point CAVE system provides high vertex count rendering of an immersive 3D environment. A standard head- and wand-tracking allows movement control and modification of the scene via the on-screen, 3D menu, while a tablet touch screen provides multiple navigation modes and a 2D menu. Graphic optimization provides theoretically limitless volumes to be presented and an on-screen mini-map allows users to quickly orientate themselves. A custom voice note-taking mechanism has been installed, allowing scenes to be described and revisited. Finally, ray-casting support allows numerous analytical features, including 3D distance and volume measurements, computation and presentation of statistics, and point-and-click retrieval and presentation of raw electron microscopy data. The extension of this system to the Unity3D platform provides a low-cost alternative to the CAVE. This allows users to visualize, explore, and annotate 3D cellular data in multiple platforms and modalities, ranging from different operating systems, different hardware platforms (e.g., tablets, PCs, or stereo head-mounted displays), to operating in an online or off-line fashion. Such approach has the potential to not only address visualization and analysis needs of neuroscientists, but also to become a tool for educational purposes, as well as for crowdsourcing upcoming needs for sheer amounts of neuronal data annotation.

Acknowledgements

I would like to acknowledge those that support me through during my ever continuing education.

Firstly, I would especially like to thank my committee chair, research advisor, and mentor, Dr. Gianfranco Doretto. Dr. Doretto has committed countless hours commenting, critiquing, and educating me on both large ideas and minute details. He continues to inspire me to constantly better myself, both academically and personally. He has shown me sides of computer science I would never had known existed, including well designed algorithms, computer vision, and machine learning. I would not be half the computer scientist today without him.

Secondly, I am also very grateful for having a terrific research host and mentor, Dr. George Spirou. Dr. Spirou has provided a wonderful environment and opportunity to lead and foster a team of undergraduates, allowing me to grow as a project leader. Especially, Dr. Adjeroth gave me lots of help during my graduate years in this department.

My sincere thanks also go to Dr. Ramana Reddy for participating on my research committee and providing one year of support as his teaching assistant. Without that experience I literally would not be where I am today, as I found this research opportunity while working in his Senior Design class.

In addition I would also like to thank my fellow lab mates for their stimulating discussions, assistance, generosity, support and their advice.

Special thanks to Erica Stewart, for all the late night support – this thesis couldn't have been written without her.

Finally, I would like to extend many thanks to my friends and my parents for their constant help and support throughout all my life.

Contents

Acknowledgements	iii
List of Figures	vi
List of Tables	viii
1 Introduction	1
1.1 Introduction	1
2 Overview of Neuronal Data Tools	4
2.1 Literature Review	4
2.1.1 Imaging Large Volumes of Neural Tissue	4
2.1.2 Neuroinformatics	4
2.1.3 Partitioning the Brain	5
2.1.4 Visualizing Modeled Neurons	6
2.1.5 Stereoscopic Vision	7
2.2 Immersive Virtual Reality	7
2.3 Virtual Reality Middleware	7
2.4 3D User Interaction	8
3 Mining 3D Nanoscale Images	9
3.1 Mining 3D Nanoscale Image Volumes	9
3.2 Visualization and Interaction Requirements	10
4 Design and Implementation	12
4.1 Design and Implementation: CAVE	12
4.1.1 Hardware System	12
4.1.2 CalVR	12
4.1.3 BrainTrek	13
4.1.4 User Interface	14
4.1.5 Cell Management	15
4.1.6 Wand Interface	16
4.1.7 Tablet Interface	16
4.1.8 Scene Modification	16
4.1.9 Navigation	17

4.1.10	Computational Representation	19
4.1.11	Loading Large Volumes	21
4.1.12	Hot Swapping	22
4.1.13	Level Of Detail	22
4.1.14	Multithreaded Loading	23
4.1.15	Filtering and Color Mapping	23
4.1.16	Special Features	24
4.2	Design and Implementation: Unity3D	27
4.2.1	Motivation	27
4.2.2	Unity and BrainTrek	27
4.2.3	ssBrainTrek	29
4.2.4	Annotation Tools	30
5	Experimental Results	33
5.1	Experiments	33
5.1.1	Neuron Resolution	33
5.1.2	Frame Rate	34
5.1.3	Cell Set Size	35
5.1.4	UI Evaluation	35
6	Conclusion	37
	References	38

List of Figures

- 1.1 **BrainTrek.** Left: A close-up view of the BrainTrek system and on-screen menu. Two cells with different cell body transparencies reveal the contained nucleus (red) with varying clarity. Right: A full view of the six 55-inch panel screens displaying nine complete cell models. The tracking system rests above the top middle panel. 2
- 4.1 **BrainTrek architecture.** CalVR is built upon OpenSceneGraph, an opensource toolkit for OpenGL. While BrainTrek leverages many innate features of CalVR (like head and wand tracking), direct modification of the core was necessary for neuron viewing. A Java-based *Mission Control* was designed for the fast management of cellular sets, and a novel tablet application was developed for controlling the system. 13
- 4.2 **Onscreen menu layout, and tablet layout.** (A) Controlled by the wand, the heads-up display menu allows for modification of the cell scene. (B) Designed for the Nexus 7, the tablet allows for modification of the scene with sliders and buttons. Icons on tablet represent, from top-left: synaptic inputs, dendrites, axon, cell body, and nucleus. Icons across middle represent: global select and de-selection of cell parts, home view, save view, folder of saved views and delete views. Buttons for cell selection described in text. Lower right: power button. 14
- 4.3 **BrainTrek data flow diagram.** After data collection, each brain volume must be properly named for processing by the system. Contour data from segmentations (in our case using IMOD) must be converted into an OBJ mesh file, and these mesh files split into individual parts and are named according to the Neuroscience Information Framework [1]. Mesh decimation is performed by quadratic edge collapse. The reduced OBJ files are loaded into *Mission Control*, which organizes and presents cell options to the user. After the user selects the cell set, *Mission Control* rewrites the hardware display system's configuration file and the scene can be launched. 14

4.5	Feature visibility, and system performance. Left: Specific neuronal parts and features were chosen to provide a metric for effects of varying resolution. Below 2 vertices/ μm^2 , even large objects like the cell body have distorted features. Above 8 vertices/ μm^2 , all features are sufficiently preserved. Center: Rendering frequency update in frames per second (fps) versus the combined size of all expressed in number of vertices in millions. Right: Rendering frequency update in fps versus the size of the cell set in the scene. Each cell model is set to the minimum resolution in vertices/ μm^2 that guarantees that all the features are visible. . . .	17
4.6	The Tablet Cube Volume Control System.	18
4.7	The Tablet Joystick Control System.	19
4.6	Comparative resolutions. Closeup view of cell body (grey) and axons that innervate the cell (colored objects). The same model is shown at various levels of mesh decimation. (A) The model at full resolution vertex density of 64 vertices/ μm^2 . All features, including fine processes called neurites (shown by white arrow) on the cell body and dendrites, are visible and retain key features of their geometry. (B) At a vertex density of 8.2 vertices/ μm^2 (86% reduction), all objects are smoothed but neurites retain their geometry. (C) At a density of 2.60 vertices/ μm^2 (96% reduction), the cell body is smooth but thinnest portion of neurites is lost. (D) At a density of 1.48 vertices/ μm^2 (D), cell body has an angled appearance and neurites are not visible.	20
4.7	The Mission Control Loading System.	21
4.8	10 Color Coded Cells.	23
4.9	81 Cells Shown Via Map.	25
4.10	Precomputed Data Displayed to User.	26
4.11	Electron Microscopy Image Projection.	27
4.12	BrainTrek on Unity.	28
4.13	BrainTrek, Regular View and Anaglyphic Mode.	29
4.14	BrainTrek Viewed With Oculus Rift.	29
4.15	User Controls for Both Systems.	30
4.16	Cell Comparison. Two cells are compared. Morphological similarity is rated on a 1 to 5 scale.	32
5.2	Tablet vs Wand. User interaction times are measured for specific tasks for both the wand and tablet. The first five tasks only involve global changes to the set of cells, while the latter five involve specific cell selection and modification.	34

List of Tables

5.1	Task list and comparative survey. Left: Each participant was educated on a specific device and asked to perform the tasks in this list. Right: Once participants had tried both input devices, they selected which device they preferred for various qualities.	35
-----	--	----

Chapter 1

Introduction

1.1 Introduction

A major goal in neuroscience is to identify neural circuits and associate them with computational functions. Many tract tracing techniques exist to provide partial identification of neural circuitry at light microscopic resolution, using tracers that are confined inside single cells or that can cross synapses, such as viral vectors [2]. These labeling techniques, although powerful, only partially reveal neural connections. Identification of complete neural circuits is best accomplished using unbiased techniques that reveal all neurons and their locations of functional contact. Currently, the only technique that accomplishes this goal without bias and with the resolution necessary to identify functional contacts is electron microscopy (EM).

Given the small dimensions of many neural structures, tracing them across serial sections requires section thicknesses of only tens of nanometers. This restriction in turn requires the assembly of hundreds to thousands of sections to construct a meaningful 3D depiction of local circuit connections. Despite these challenges, the landmark assemblage of the entire circuit map, or connectome, of a species was accomplished nearly 30 years ago for the nematode *Caenorhabditis elegans*. This effort required over 8,000 serial sections and more than a decade to collect and analyze the EM images [3] and presaged mapping of the human genome, which likewise occupied more than a decade. The human genome project provided baseline information that has underpinned many advances in biomedical research and spawned new technological developments and methods for data analysis.

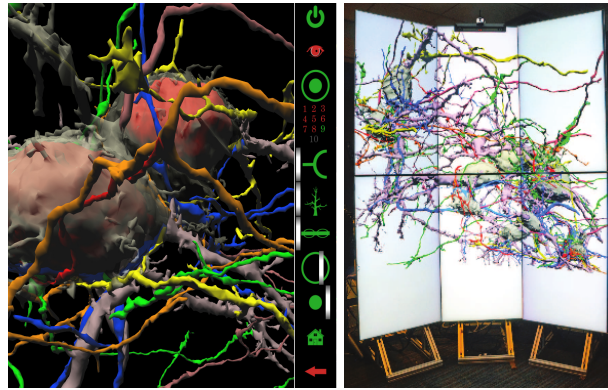


Figure 1.1: **BrainTrek**. Left: A close-up view of the BrainTrek system and on-screen menu. Two cells with different cell body transparencies reveal the contained nucleus (red) with varying clarity. Right: A full view of the six 55-inch panel screens displaying nine complete cell models. The tracking system rests above the top middle panel.

Parallels between the human genome project and mapping brain connectomes at high resolution are striking. Technical advances over the past decade have increased the speed of acquiring serial section EM images (ssEM). New, semi-automated strategies to extract neural connection diagrams have been developed and improvement is constant. Several groups, ours included, have realized the value of these image volumes beyond assembling wiring diagrams and thus investigate the 3-dimensional structure of brain cells and their spatial relationships. From this approach we hope to appreciate not only functional contacts between neurons, but also among neurons and glia, and gain insight into higher order organizational features of brain tissue. These features can include organization of fiber fascicles, geographic distributions of neurons relative to glia, and orientations of dendrites relative to neural inputs.

The resulting wealth of information generated by segmenting objects out of ssEM image volumes requires its own analysis pipeline to identify data trends. These data sets offer opportunities to study novel aspects of 3D brain structure, for which analysis tools may not exist. We suggest that visualization of complex brain structure is a useful first step to gain initial insight into general features of organization that can be re-phrased as hypotheses for testing by subsequent quantitative analysis.

Reconstruction of all cells in a given EM image field has great power, but when viewed in total is as undecipherable as the block of tissue prior to sectioning and staining. Of key importance for visualization is the ability to place an understandable number of cells into a field of view and then manipulate the viewscape efficiently. This goal carries significant informatics challenges.

The mapping of neuron contours into objects at nanoscale resolution creates cell models that are computationally expensive to manipulate. Each neuronal model can be on the order of 250,000 vertices, so rendering large numbers of these models in a cost-efficient manner is a primary challenge. Furthermore, the capabilities of viewers to comprehend a busy visual scene must be considered. Complexity of the visual scene can vary as the view, or the types of objects displayed change, and by perception of the viewer. Therefore, a user interface must be developed that performs operations to filter or alter the display to suit the scientific and psychological needs of a range of viewers. This user interface (UI) should be intuitive and therefore trade flexibility with simplicity.

In this work I develop and evaluate BrainTrek (see Figure 1.1), a 3D immersive virtual reality (IVR) environment, which aims at addressing the challenges expressed above. In particular, the system includes:

- A 3D IVR system for displaying high-resolution 3D cell models extracted from ssEM image volumes of nanoscale neuronal tissue;
- A 3D user interface (UI) for the interaction with and manipulation of spatially calibrated neuronal models;
- A management system for accessing and browsing collections of brain tissue volumes through developmental ages and space;
- A pre-processing pipeline for optimizing the representation of neuronal data while balancing performance and visualization accuracy;
- An enhanced access and control of system features through a combined use of a wireless tablet and a wand;
- A UI evaluation study performed with neuroscientist users.
- A Unity3D port developed to further increase usability of the system.

Chapter 2

Overview of Neuronal Data Tools

2.1 Literature Review

2.1.1 Imaging Large Volumes of Neural Tissue

New, big data electron microscopy techniques have changed the approach to 3D imaging. These techniques employ high throughput transmission EM (TEM) approaches, such as TEM camera array (TEMCA), or scanning EM (SEM) approaches, such as serial blockface SEM (SBEM), automated tape ultramicrotome (ATUM), or focused ion beam scanning EM (FIBSEM) [4]. Image volumes are smaller than those collected using light microscopy, but can exceed millions of cubic micrometers [5, 6, 7, 8] with increases of several orders of magnitude on the horizon. These image volumes pose new opportunities to segment individual neurons into their major components (e.g. cell body, axons, dendrites), subcellular organelles (e.g. mitochondria, endoplasmic reticulum) [9], and assemble neurons into circuits. Accumulation of these segmented objects places new demands for flexible systems to visualize combinations of these objects that cross the scale from organelle to circuit [10].

2.1.2 Neuroinformatics

Presently, image collection is not the limiting factor for nanoscale connectomics. Modern imaging technology can collect high resolution images at a rate that far outpaces current ability to segment and analyze tissue elements. Image volumes must be both archived and accessible

for ongoing analysis, so storage and transmission of image data becomes a serious challenge. As examples, Bock et al. [5] amassed 36 terabytes of raw data using the TEMCA system that was ultimately reduced to a size of about 10TB ($450 \times 350 \times 52\mu\text{m}$) after stitching tiles together. Nanoscale connectomic information on the mouse retina [6, 8], was extracted from image volumes up to 2TB ($300 \times 350 \times 60\mu\text{m}$). Our data volumes approached 1.5TB ($125 \times 96 \times 80\mu\text{m}$) but volumes at multiple developmental ages were collected for a total of 6TB [7]. Work is conducted at two sites: images are collected and archived at one site (UCSD), but communicated to and analyzed at a different site (WVU), where backup copies must be archived. This mode of operation will become more prevalent as collaboration increases among distant groups.

Segmentation of objects from image volumes generates new large sets of data. In our procedures, the collected models for single cells (cell body, dendrites, axon, nucleus, afferent axons and nerve terminals) can be as large as 80MB. Inclusion of additional subcellular organelles could increase this file size by an estimated factor of 2-3. Image volumes of 1,000 cells could yield model files that in total reach a size of 250GB. These file sizes do not currently test storage requirements but will do so as the number of reconstructed cells scales upward. Current file sizes, however, do pose challenges for manipulation and display in a virtual reality (VR) environment. A key requirement to operate with these data in VR is the means to organize cellular models and select them to generate useful viewscales for exploration.

2.1.3 Partitioning the Brain

Essentially two approaches are taken to extract information from these image volumes. The first is to map neural circuitry, which requires identification of neural processes and can be represented as skeletonized structures with identification of functional contacts between neurons. The second is to provide accurate 3D representations of structures, which requires their identification followed by segmentation by outlining enclosing membranes and tracing non membrane-enclosed structures such as cytoskeletal filaments. Both skeletonized and volumetric approaches benefit from methods to segment membranes, but our focus here is on generating 3D geometric representations.

Even moderately-sized EM image volumes ($1 \times 10^6 \mu\text{m}^3$) may contain on the order of 10^2 neu-

rons and glia, 10^3 cellular parts, and 10^6 subcellular organelles. Densely packed neural territories such as retina contain an order of magnitude greater number of cells in a similar volume [8]. These values scale by 10^3 for an image volume of 1mm^3 , which is still insufficient to capture entire cell groups in small mammals such as mice. Therefore, a premium is placed upon automated segmentation approaches. Current technology for segmentation and annotation of cellular parts relies primarily upon manual methods, scalable by recruitment of participants within institutions [7, 6], online recruitment through game participation [11], or financial compensation (UCSD). Although manual techniques are tedious and slow, concerted effort can accumulate a large database of segmented cells. A recent report using specialized staining techniques that augment cell membranes, and amenable to semiautomated approaches, yielded volume reconstruction of about 1000 cells [8].

2.1.4 Visualizing Modeled Neurons

3D rendering of large-scale data sets permits analysis of cellular features with greater statistical reliability due to the large number of cells. Examples include diameters and surface areas of cellular parts, locations and sizes of synaptic contacts, and quantification of these features relative to internal organization of the brain region (e.g. cell layers) or orientation of the cell group relative to cardinal orientation planes (e.g. medial-lateral and dorsal-ventral). The resultant image volumes also provide opportunities to explore and identify novel features of cellular organization that can best be discovered by exploration of the 3D structure. Led by data-intensive fields outside of neuroscience, such as GIS, scientists increasingly rely on 3D visualization to grapple with questions of form and function in their data [12].

Display of 3D can be made in 2D on computer screens, but also in VR systems described as look-in, as in 3D television or movies where the viewer remains outside of the object field, or immersive (IVR). In scientific applications, IVR users report better insight into spatial relationships [13, 14], including neuroscience applications such as viewing white matter tracts revealed by diffusion tensor imaging [15]. IVR imparts a sense of presence in the scene through a wider field of view that recruits peripheral vision, and refinements that incorporate aural and haptic sensoria are underway [16, 17]. Several studies compared look-in and IVR; users perform certain searches

more efficiently in IVR, but tend to prefer whichever system offers higher brightness and spatial acuity [18, 19, 20].

2.1.5 Stereoscopic Vision

Stereoscopic projection is a technique used to create the illusion of depth. Essentially, stereoscopic visualization is commonly achieved by presenting slightly different images to each eye. The different images are merged via human stereopsis. This was first explained as early as 1838, when Charles Wheatstone, inventor of the stereoscope, wrote, “the mind perceives an object of three dimensions by means of the two dissimilar pictures projected by it on the two retin” [21].

2.2 Immersive Virtual Reality

Developed in 1992, the CAVE (cave automatic virtual environment) is a fully immersive virtual reality environment. It utilizes image projection onto multiple walls or screens via forward and back projection. Head and body tracking is possible for perspective enhancement [22]. In very recent years the availability of inexpensive LED-TV panels combined with polarized 3D glasses have enabled a new generation of low-cost CAVEs [23].

Forsberg et al. [24] compared the exploration of laser confocal microscopy data by biologists using either a standard desktop computer, a fishtank virtual reality system, or a CAVE. Fishtank VR is a standard desktop monitor equipped with stereoscopic projection. They found the biologists’ performance improved directly with immersivity, with significant gains in understanding spatial relationships and identifying important features. Furthermore, biologists uniformly preferred the CAVE as a means of studying 3D data. With the increase of interdisciplinary research and massive datasets, several similar calls for large data visualization have been made [25, 26].

2.3 Virtual Reality Middleware

Software for driving CAVEs is both numerous and varied in functionality. Commercial products like CONVERSE and AVANGO do allow the creation of custom applications, but do not provide source code and are thus unmodifiable at the core level [27, 28]. Various open source systems

provide an extensible framework, but may lack important capabilities like multi-screen displays, multiple users, cluster-awareness, and multi-GPU capability. CalVR [29], a novel virtual reality middleware, provides all of these features along with higher order capabilities like fully customizable plugins, higher-order abstraction, and a real time collaborative session support. Due to its object-oriented C++ design, CalVR allows the leveraging of pre built classes for handling screen configuration, navigation, and user interaction. Although we do not leverage the cluster support for networked PCs, further expansion of the systems rendering capability could easily be accomplished by the addition of more PCs.

2.4 3D User Interaction

Interfaces for system control tasks in virtual environments have been extensively studied. Various types of interfaces have been analyzed for speed, error rate, and number of control actions. Qualitative studies on learning rates, ease of use, frustration, and perceived difficulty of the performed task have also been undertaken [30, 31]. Investigations into bimanual interaction, proprioception, and passive-haptic feedback have been considered [32]. Input devices such as PDA, token, gesture, and voice interaction techniques each have various strengths and weaknesses, and have been quantifiably measured [33]. Since user interaction is often context-sensitive, a unique system is needed for the presentation of neurons.

Chapter 3

Mining 3D Nanoscale Images

3.1 Mining 3D Nanoscale Image Volumes

Brains are constructed via complex events that regulate formation of neuron groups from specified precursors, migration to appropriate regions of the brain, extension of axons to target cell groups and genesis of specific loci of functional contact. These events are subject to evolutionary selection. Neurons function to encode information by integration of their inputs. The timing of inputs can grossly affect resultant activity patterns, so it is likely that factors that affect timing are among those which exert evolutionary pressure on brain architecture. These factors can include cell group location, separation distance between communicating cell groups and axon diameter, along with other factors that determine action potential conduction velocity. These factors in turn are influenced by the geometric constraints for packaging all brain elements inside of its bony enclosure. There is much to be learned, then, from the 3D geometry of rendered brain tissue, which complements the goal of connectomics to extract circuit diagrams.

Neuronal arrangements are not crystalline, especially outside of layered regions such as retina and cortical tissue, but neither are they amorphous. Collections of cells, called nuclei, may possess orientations of cell bodies and dendrites that show general or regional trends that are made less regular by the need to incorporate other structures into the local volume such as the vascular network or bypassing axon bundles. These cellular arrangements in turn may complement the organization of afferent fiber fascicles, for example, and the pattern of emergence of inputs from these fiber groups. The appropriate ways to quantify these spatial arrangements are not all obvious.

A method that promises efficiency for analysis is to inspect the segmented cell volume using IVR, and use these derived insights as the basis for new analysis strategies to quantify the observations.

One goal of connectomics is to compare wiring diagrams of healthy vs pathological tissue. Network analysis approaches of complex circuit diagrams can likely reveal many key features of pathology. 3D rendering of segmented cells and IVR exploration promises to yield additional insights, which may reveal the signatures of aberration programs and clues into disease mechanisms. In summary, many fundamental questions are unanswered about brain 3D organization which IVR can help address.

3.2 Visualization and Interaction Requirements

For a single large image volume, one thousand cells or more may be present. If these cells are all traced, separated into parts, and modeled, the total file count could reach above ten thousand files. This causes a strong need for file organization and authentication.

The study of neurogenesis requires collection of multiple temporally-related datasets. Furthermore, comparisons between different individuals or species require quickly relating similar datasets. Once volume sets have been organized and processed for viewing, users should be able to quickly select and view a specific set. A volume set hierarchy should be established and an easy-to-use selection interface should be developed.

Despite advanced computer graphics rendering, polygon count still limits the ability to smoothly present large numbers of models. The IVR system must be capable of visualizing many neurons in spatial relation without lag or delay. The system should be able to present a set of cells at greater than 10 frames per second (industry minimum for fps). If lower resolution models are required to achieve usable performance and large data volumes, a balance between rendering performance and model accuracy is essential.

Having created and prepared the three-dimensional cells, an effective method to view them is required. This method would need to have the capability to render large quantities of cell models while maintaining their spatial relationship. Furthermore, once the cells are rendered, they must be manipulated to allow key observations to be made. A method of operating the scene so that cell sets could be viewed from multiple angles is needed.

Once rendered, these large sets of cells can quickly become overwhelming to the user. They must be presented in a fashion that aids visual comprehension and allows manipulation of the scene. The ability to alter the opacity of whole cells or parts of cells provides a means to observe structures in relation to otherwise obstructing structures. Intuitive control of the viewing perspective is vital. The ability to save these perspectives would be useful in collaboration and reporting.

Chapter 4

Design and Implementation

4.1 Design and Implementation: CAVE

4.1.1 Hardware System

Our CAVE system is comprised of six 55-inch 3D enabled LCD/LED screens (LG 55LM9600) and follows the TourCAVE design developed and tested at Calit2 [34]. We employ an ART (Advanced Realtime Tracking) SmartTrack system for head and wand tracking, which works in a small-volume (2m^3) and is based on an infrared optical tracking camera. It utilizes small infrared reflecting spheres to track head and wand location and orientation. CalVR also has basic support for the Microsoft Kinect, making that a viable option at an even lower cost. The computer driving the CAVE contains an Intel Core i7-3820 processor, two GeForce GTX 680 GPU cards, and 32GB of RAM. The machine is running CentOS 6.3. A Nexus 7 tablet provides an alternative user interface. This system provides comfortable viewing by two people.

4.1.2 CalVR

CalVR provides the middleware framework that drives BrainTrek. Developed for CentOS, CalVR is a C++ object-oriented class hierarchy that utilizes the OpenSceneGraph library and OpenGL to render graphic output. It supports multiple 3D menu systems, multiple navigation methods, and multiple tracking and display systems. It possesses a plugin system for the development and integration of custom applications without needing to compile the core system. The

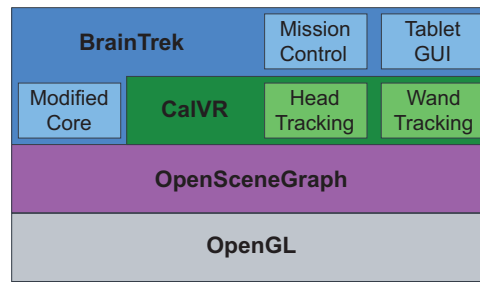


Figure 4.1: **BrainTrek architecture.** CalVR is built upon OpenSceneGraph, an opensource toolkit for OpenGL. While BrainTrek leverages many innate features of CalVR (like head and wand tracking), direct modification of the core was necessary for neuron viewing. A Java-based *Mission Control* was designed for the fast management of cellular sets, and a novel tablet application was developed for controlling the system.

configuration of CalVR can be modified by editing an XML file; this feature provides a location system to save various configurations, including cell sets and viewing system data, for later viewing. Object classes describing the 3D menu system allow for complete customization, which we leveraged to optimize the user interaction for neuron viewing.

Since rendering large scenes of complicated object models is computationally expensive, there are a number of software techniques to improve performance. Backface culling, degrees of interest, and mesh decimation can all be utilized to achieve desired levels of performance. CalVR comes equipped with many useful features for rendering, including a unique culling visitor system.

4.1.3 BrainTrek

BrainTrek is a plugin-enhanced and core-altered modification of CalVR for the specific use of viewing multi-layered neuron models (see Figure 4.1 for the architecture overview). Its name reflects the user experience when cells are enlarged to human dimension and navigated as one would hike through the backcountry of boulders, trees, and other life-sized objects. The experience is reminiscent of the concept behind “Fantastic Voyage,” a movie in which humans were reduced to cellular dimensions (see Figure 1.1). Although BrainTrek leverages a large amount of the core CalVR functionality, modifications of the core kernel have been made to further specialize the system for a specific neuroscience task. In the following sections, we will discuss several features in detail.

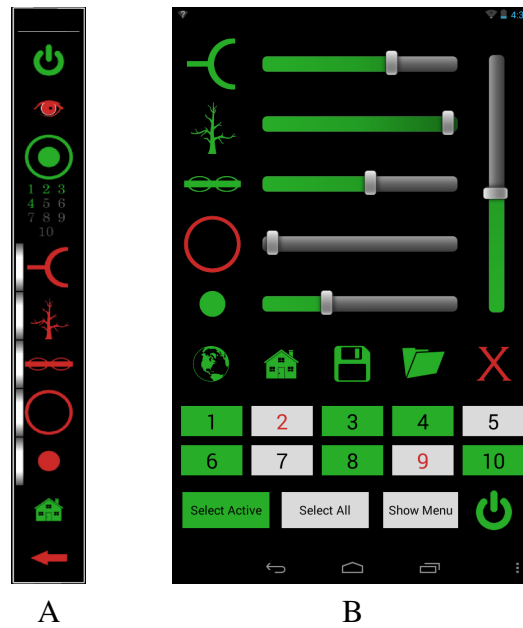


Figure 4.2: **Onscreen menu layout, and tablet layout.** (A) Controlled by the wand, the heads-up display menu allows for modification of the cell scene. (B) Designed for the Nexus 7, the tablet allows for modification of the scene with sliders and buttons. Icons on tablet represent, from top-left: synaptic inputs, dendrites, axon, cell body, and nucleus. Icons across middle represent: global select and de-selection of cell parts, home view, save view, folder of saved views and delete views. Buttons for cell selection described in text. Lower right: power button.

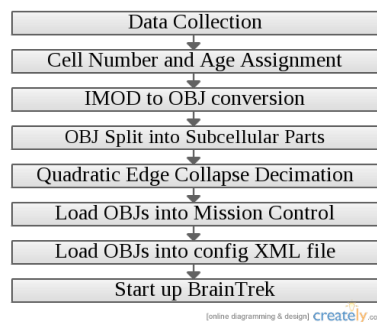


Figure 4.3: **BrainTrek data flow diagram.** After data collection, each brain volume must be properly named for processing by the system. Contour data from segmentations (in our case using IMOD) must be converted into an OBJ mesh file, and these mesh files split into individual parts and are named according to the Neuroscience Information Framework [1]. Mesh decimation is performed by quadratic edge collapse. The reduced OBJ files are loaded into *Mission Control*, which organizes and presents cell options to the user. After the user selects the cell set, *Mission Control* rewrites the hardware display system's configuration file and the scene can be launched.

4.1.4 User Interface

The user interface occurs by using the ART system and a Gyration 3D Mouse to track the wand and its interaction with an on-screen menu or via a tablet (see Figure 4.2). In both instances, our

goal was to design an interface that was aesthetically pleasing, by keeping its look sharp, clean, and simple while maintaining speed and efficient functionality. A sharp and clean look was achieved by choosing a color scheme of very bright green and red icons and buttons on a black background. Inspired by Google Body Browser [35], each cell part (body, nucleus, dendrite, input, axon) was assigned a dedicated icon. An icon based menu was developed to facilitate ease of recognition and approachability for novice users [36]. These icons were carefully selected to represent their respective part intuitively while maintaining simplicity. Users can interact in an easy, satisfying way with the UI to adjust the set of viewed objects by using the sliding controls. Every cell part has a dedicated slider to control transparency (see Figure 4.2a). A main slider can be used to bring a cell into the picture one part at a time, creating the feeling of moving through layers. For drastic resets to the set view, a `Globe` function allows instant alteration between full opacity and full transparency on all parts.

4.1.5 Cell Management

Currently, the interface allows for the independent control of up to 10 cells. Each cell is represented by a number ID, and each ID button can be pressed to “select” the cell. Selected cell buttons are highlighted in green. Once a cell is selected, manipulation of the transparency sliders will affect it. This allows for rapid selection and alteration of individual or sets of cells. In preliminary testing, however, neuroscientists often found certain cells in a set unnecessary to their specific inquiry. The ability to drop an entire cell entirely from the scene was requested and was implemented using long-hold functionality of the tablet. If one of the cell buttons is held for an extended moment, a full transparency is applied to that cell. Removal of the cell is indicated by changing text in the cell button to red, and an automatic de-selection of the button, indicated by turning the button white. This red text describes an inactive cell. To facilitate even faster cell selection, two buttons, `SelectActive` and `SelectAll` were implemented. `SelectAll` simply selects or deselects all of the cells. This assists quickly switching between modifying the entire set and one individual cell. `SelectActive` is used for the same purpose, but it will only select active cells.

4.1.6 Wand Interface

The BrainTrek system is equipped with an ART (Advanced Realtime Tracking) SmartTrack system for head and wand tracking, which works in a small-volume ($2m^3$) and is based on an infrared optical tracking camera. It utilizes small infrared reflecting spheres to track head and wand location and orientation. The wand allows for rapid navigation with six degrees of freedom, and manipulation of the three-dimensional menu system. The wand can be used to aim a computer generated ray-cast, allowing users to target cell models for coordinate-specific actions.

The on-screen menu can be controlled by the wand by aiming it at menu items and toggling basic options, such as cell selection. Transparency sliders are moved by selecting and dragging within the menu. A cell selection heads-up display indicates which cells are currently selected and thus alterable. The wand can also be used to reposition the menu bar in VR space. A user can fly through the model by twisting, tilting and dragging the wand.

Although the wand is versatile and useful, it can sometime be difficult to use, especially for new users. For example, often users will want to rotate the camera around a specific cell, creating the illusion that the cell is spinning in place. With the wands six degrees of freedom, experienced users can achieve this by combining multiple gestures, but the control system is not always intuitive for first-time users. Since many users include visiting researchers who will only have a short opportunity to study the system, the development of alternative control systems was crucial. A novel scene manipulation system, as well as two alternative navigation systems have been developed, both utilizing the tablet.

4.1.7 Tablet Interface

4.1.8 Scene Modification

While the wand is integral to scene navigation, we noticed early on it is difficult to use with the menu system. Due to the heavy use of sliders, fine manipulation of the interface was unwieldy due to the sensitivity of the wand. An Android tablet with a custom interface was developed to give the user an alternative to the wand. The tablet connects to the system through the CalVR socket communication system. Many benefits were gained from having a tablet control features to our

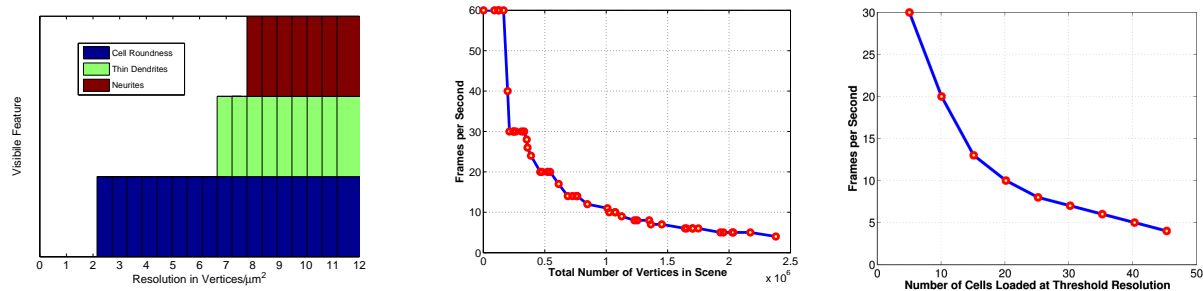


Figure 4.5: **Feature visibility, and system performance.** Left: Specific neuronal parts and features were chosen to provide a metric for effects of varying resolution. Below 2 vertices/ μm^2 , even large objects like the cell body have distorted features. Above 8 vertices/ μm^2 , all features are sufficiently preserved. Center: Rendering frequency update in frames per second (fps) versus the combined size of all expressed in number of vertices in millions. Right: Rendering frequency update in fps versus the size of the cell set in the scene. Each cell model is set to the minimum resolution in vertices/ μm^2 that guarantees that all the features are visible.

system. By utilizing the entire tablet screen, a spacious layout was possible, allowing ease of use (see Figure 4.2b). The touch screen capabilities of the tablet translated well into implementation of sliders from our UI. Furthermore, more advanced controls could be developed on the tablet that were not easy via the wand-control system. Selection speed of groups of cells is drastically increased. Touch-sensitive techniques like long-holds are possible for programming additional functions without altering the design layout. For some tablet models, haptic feedback can provide users confirmation of a change. Tablets provide a space for virtual keyboards, making user input for saving data possible. Finally, voice recognition could be integrated for the addition of voice commands.

4.1.9 Navigation

Cube Volume

The Cube Volume control system is designed to be subject-centric; it allows users to focus on a specific cell model and examine it from every angle. The concept is simple: on the tablet touch screen, a three dimensional cube is presented. This cube represents the bounding box of the

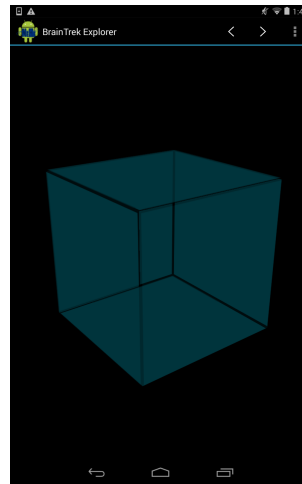


Figure 4.6: **The Tablet Cube Volume Control System.**

neuronal volume presented on the BrainTrek system. Users may utilize common swiping gestures on the touch screen to rotate the cube, and pinch-to-zoom gestures to increase or decrease the size of the cube. These changes are reflected exactly on the BrainTrek volume. To increase the subject-centricity, the viewing volume is automatically centered on a cell. This is done by performing a translation on the volume such that the center of the cell is realigned to the origin coordinates. The selected center cell can be easily changed via arrow keys, and the color-coding system allows users to pinpoint which cell they would like to center. With the Cube Volume control system, performing perfectly spherical orbits around a cell model is intuitive and simple. Because of the prevalent usage of smart phones and tablets, many users have already been introduced to these gestures and are familiar with their uses.

Virtual Joysticks

The second control system is subject independent and is designed to replicate the wands navigation functionality. To achieve this, a virtual joystick was created for the touch screen tablet. Two large circles were drawn to represent joystick areas. Touching the screen within the joystick areas results in a vector and magnitude calculation; this is reflected by moving the camera through the neuronal volume. The magnitude calculation is polynomial; as users touch the area at longer diameters, the magnitude calculated grows faster. This allows fine manipulation at short distances and large movements at long distances. The left joystick area controls translational movement,

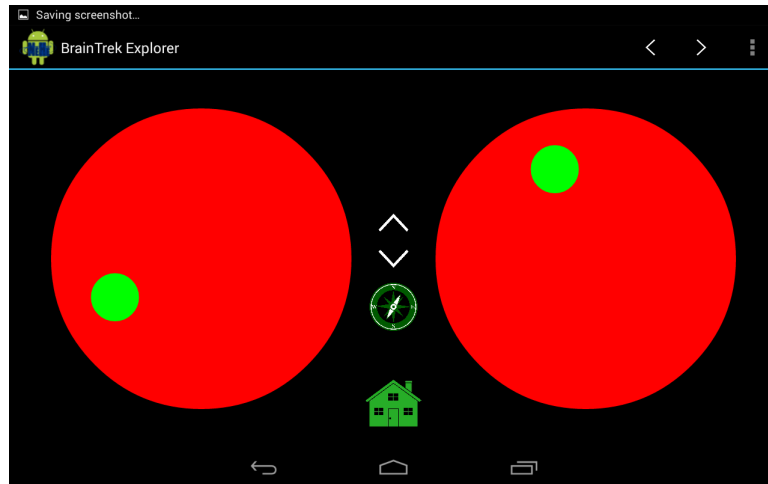


Figure 4.7: **The Tablet Joystick Control System.**

while the right joystick controls the viewing angle. Two center arrows allow for altitude changes. The three control areas give six degrees of freedom, allowing the virtual joystick to replace the wand if needed. Users may quickly toggle the Map functionality via the Compass icon.

4.1.10 Computational Representation

BrainTrek supports a coded hierarchy for describing each model object. Cells from the same image volume are loaded together as a “set”. Cells of a set retain their spatial relationships, allowing the formulation of structure-based insights, which we term hypotheses. Hypotheses deemed to have substantial merit can be tested by development of new analysis tools offline. Every cell is composed of individual models representing anatomical subsections – these subsections are organized with a part-type tag. A part-type tag simply labels an object file as a specific subpart of a neuron. For example, a model’s axons would be labeled AXON, and graphical filtering can be applied to all AXON models. This feature allows a neuroscientist to alter a scene flexibly. Part-types can be color coded for unique color schemes or highlighting. Cells are indexed and “selectable” by wand or tablet. When a subset is “selected”, users may apply various graphical effects to that subset of cells. Therefore, the user can simultaneously alter the visibility of all models comprising a single cell or a selected subset of models for that cell. Flexibility is introduced whereby the same procedures can be applied to a subgroup of cells, specific Part-types of all cells or a selected subset of cells, or different part-types among multiple selected subsets of cells.

Segmentations of individual objects from raw image data are saved as individual model files, but linked together as parts of particular cells. These model files follow a naming convention and processing pipeline prior to display in the CAVE (see Figure 4.3). Each cell is assigned an experimental identifier. The image volumes used as our testbed were collected across developmental age in the mouse, so the identifier is the age of the animal (e.g. postnatal day 3 represented as P3). Each cell in the volume is numbered and differentiated as a neuron or glial cell and assigned a cell number (`cell_glia_1`). Individual objects that are parts of these cells are named according to the Neuro Informatics Framework (e.g. `cell_neuron_1_cellbody`) [1]. These files were segmented using IMOD [37], so each file must be converted to a Wavefront OBJ file, which is performed using the `imod2obj` tool. A custom Python script analyzes and splits each OBJ file into subcellular parts and stores them in a generated subfolder directory. The standard nomenclature allows the system to load each file under the correct part type for scene manipulation.

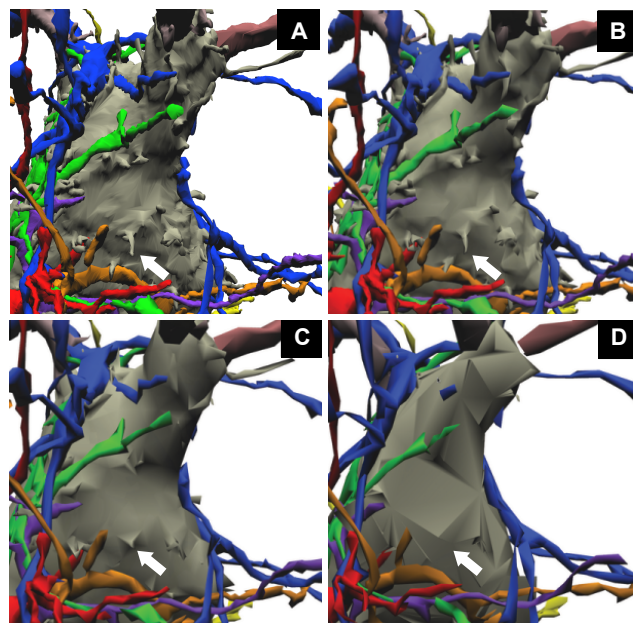


Figure 4.6: **Comparative resolutions.** Closeup view of cell body (grey) and axons that innervate the cell (colored objects). The same model is shown at various levels of mesh decimation. (A) The model at full resolution vertex density of $64 \text{ vertices}/\mu\text{m}^2$. All features, including fine processes called neurites (shown by white arrow) on the cell body and dendrites, are visible and retain key features of their geometry. (B) At a vertex density of $8.2 \text{ vertices}/\mu\text{m}^2$ (86% reduction), all objects are smoothed but neurites retain their geometry. (C) At a density of $2.60 \text{ vertices}/\mu\text{m}^2$ (96% reduction), the cell body is smooth but thinnest portion of neurites is lost. (D) At a density of $1.48 \text{ vertices}/\mu\text{m}^2$ (D), cell body has an angled appearance and neurites are not visible.

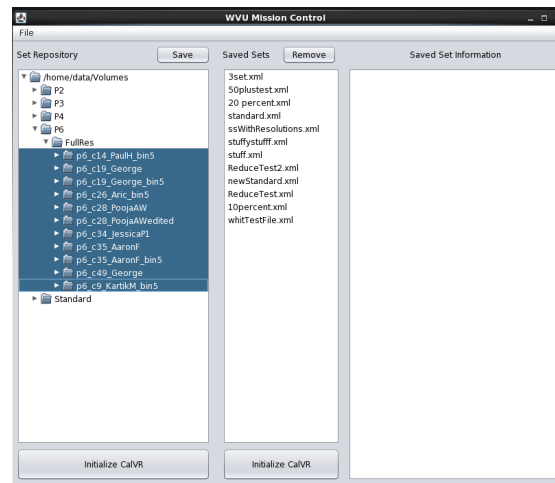


Figure 4.7: The Mission Control Loading System.

4.1.11 Loading Large Volumes

Mesh Decimation

After part creation, an optional step of mesh decimation may take place. Mesh point decimation is a common technique used to eliminate unneeded data points from a model [38]. Since frame rates and rendering time in the IVR environment is proportional to the number of vertices rendered, this step can dramatically improve performance. The open source application MeshLab is utilized to lower the overall polygon count and increase the frame rate. Specifically, the quadratic edge collapse decimation algorithm quickly and efficiently reduces the polygon count of the models. This step is likely obligatory as the number of objects in the view increases. Performance improvement by mesh decimation comes at a cost of model accuracy — a fraction of the surface features of the cells will be smoothed out by the algorithm. Due to the detailed nature of these models, a careful balance of performance and accuracy must be achieved (see Section 6.3).

Mission Control

The CalVR software has no GUI-based ability to change which 3D neuron models have been loaded into the program. An XML file instructs the system as to which 3D model files to load. Manually editing this XML file in order to change the 3D model files is inconvenient and potentially hazardous to the system: a single typo might prevent CalVR from successfully loading. A

quick and intuitive cell selection method should be in place for rapid switching of cell sets, so a JAVA-based GUI was developed. We created the *Mission Control* user interface tool to bridge the gap between the CalVR system and our volumes of 3D model files. Mission Control is a light-weight custom application for the indexing, selection, and loading of cellular model structures. Individual cells are organized by postnatal day, and indexed by assigned number. Users may select a cell and examine the subdirectory contents, allowing quick investigation of cellular parts. Once a set of cells from within a postnatal day are selected, an automated Python script rewrites the configuration XML file. This set may be viewed immediately, or saved for later viewing. The user can also view his previously saved selections, examine its list of associated files and read a brief description of the selection. However, the data must first be manually organized on the hard drive in order for Mission Control to display it properly.

4.1.12 Hot Swapping

The cave now has the ability to hot-swap the scene among configured sets of cells referred to as Structures. This opens up the possibility of exploring different ages of cells from the same area of the brain in order to inspect changes that occur at different stages of development. This was accomplished by describing, in the configuration file, labeled structures each of which list the model files that need to be loaded to display the volume. Each Structure element is then used to construct a corresponding object which handles the loading of the model files into memory when needed, and controlling the material properties of the model.

4.1.13 Level Of Detail

It used to be the case that the cave would only be capable of loading and displaying the cell models at their full resolution, but the amount of cells in a volume can exceed 80 cells. Through the use of multi-threaded loading and level of detail (LOD) simplification the cave can now display unlimited quantity and complexity of cells.

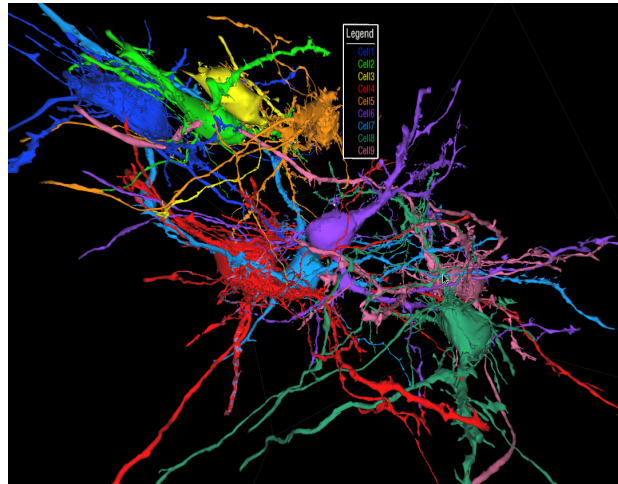


Figure 4.8: **10 Color Coded Cells.**

4.1.14 Multithreaded Loading

The system now uses a configurable number of threads to retrieve cells from a job pool and load them, we found on average a 4 fold reduction in loading times using this method. Each cell part is also now suspended in a Level of Detail adjuster which decreases the amount of detail on the screen proportional to the size of the model on the screen. This allows for fine detail to still be displayed when close to a part of a cell while preventing cells outside of regions of interest to be rendered, improving frame rate to a manageable level.

4.1.15 Filtering and Color Mapping

While this allowed for larger cell fields in the cave, it also added the problem of controlling, and identifying cells in such a large field. The solutions were the ability to filter the cells down to ten based on their proximity to the camera while making the remainder invisible, allowing for sensitive control of the cells. The user needs only to unfilter the set to bring back the entire cell field. Once filtering is done, we can also introduce color coding which overrides the colors attributed to the type of part in the cell with a color from a displayed legend which will differentiate the cells.

4.1.16 Special Features

Location Recording and Voice Notes

Due to the potentially large volumes being explored and the similarity between cell models, users will often notice a specific shape or pattern during a session only to be unable to find that pattern again. This severely dampens the ability of researchers to revisit or share findings with others. To prevent time lost by searching, and to facilitate collaboration, a location recording system was designed. At any time during the systems use, a user may press the Save Location icon on the tablet. Internally, the system serializes the camera position and angle of attack, as well as the transparency values of each cellular subpart. This information can then be easily recalled by utilizing the Load Location icon button. During initial testing, the location recorded was found to be useful, but without a descriptive note of the view, users found it difficult to remember the exact importance of a given view. To further enhance the usefulness of this system, users may now record voice notes via the tablet microphone. When creating a location note, users are asked to provide their name and a short description of the location. Then, they are given the option to record a voice note. This note can last up to five minutes, although most users only record for around thirty seconds. Upon loading a voice location, users may playback the voice note.

Map

As neuronal segmentation techniques improve, larger and larger volumes will be constructed. Even in the currently sized volumes, if the view is sufficiently zoomed into a region, users can lose their orientation. To prevent this, an on-screen map has been provided. Users can quickly toggle between the Map and regular view via the tablet Compass icon. Internally, the map is actually an exact copy of the cell models scaled down. A bounding box has been placed around the map, allowing users to understand depth. A blinking red and green dot represents the users current location.

Coordinate Specific Functions

Sometimes, users require information that is location dependent. Allowing users to aim at a specific cell model and select it provides a useful function that can be leveraged in several ways.

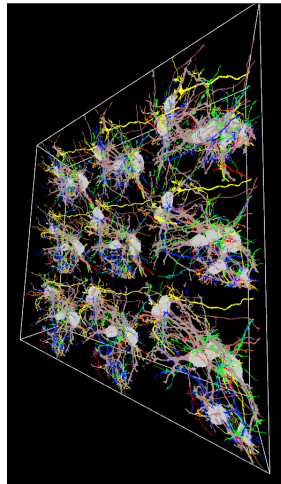


Figure 4.9: **81 Cells Shown Via Map.**

Once the coordinate is known, calculations can be made and information can be retrieved from databases.

Distance Measurements

One of the most obvious features derived from known coordinates is the distance measurement. Often researchers desire to compare two cell bodies or neurons, but are usually forced to use a 3D modeling software, like Blender or RhinoCAD. Since switching to another software program is time consuming while exploring, a simple measurement system is very useful. Users are provided two distance markers (D1 and D2) via the on-screen cursor menu. If both of these distance markers are placed into the volume, a Euclidian distance is automatically calculated and presented to the user. The distance marker system is ideal for neuronal models, as users may desire to track a nonlinear structure, such as a dendrite, along the course of its path. By hopping distance markers along the path and summing the reported distances, users can retrieve the total distance with a minimum of mouse clicks.

Statistic Viewing

Despite the ease of collecting distance measurement, not all quantifiable measurements are easily possible in a three dimensional environment. More complicated statistics like surface area and volume are extremely difficult for users to specify. To allow some amount of information to

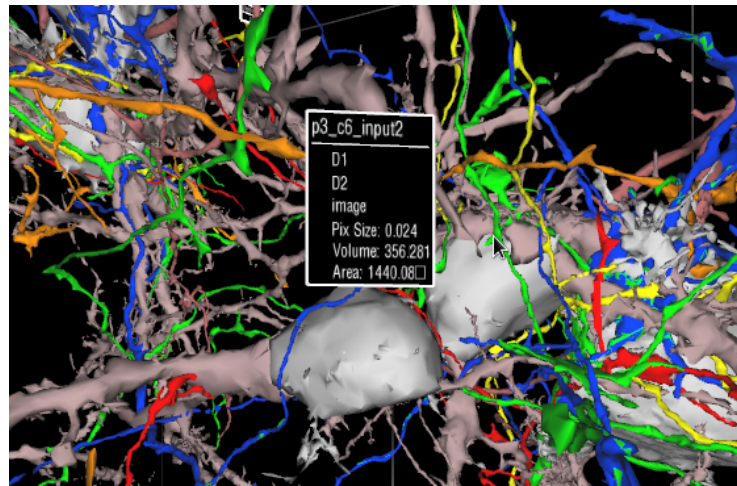


Figure 4.10: **Precomputed Data Displayed to User.**

the user, a database of model volumes and surface areas was constructed. By specifying a model, a user is presented with the volume and surface area. If the model is an input, and the synapse area is known, that figure is also presented.

Electron Microscopy Projection

Since the neuronal models are all extracted from electron microscopy, it is often useful to view the section of EM that the specific model came from. With coordinate specific ray-casting, users may select a coordinate via the wand pointer and project the EM image onto the screen. This provides a deeper understanding, especially for unsegmented neuronal structures like synaptic vesicles. In future work, expansion of this system should include the ability to display the exact coordinate of the vertex, and allow users to zoom in.

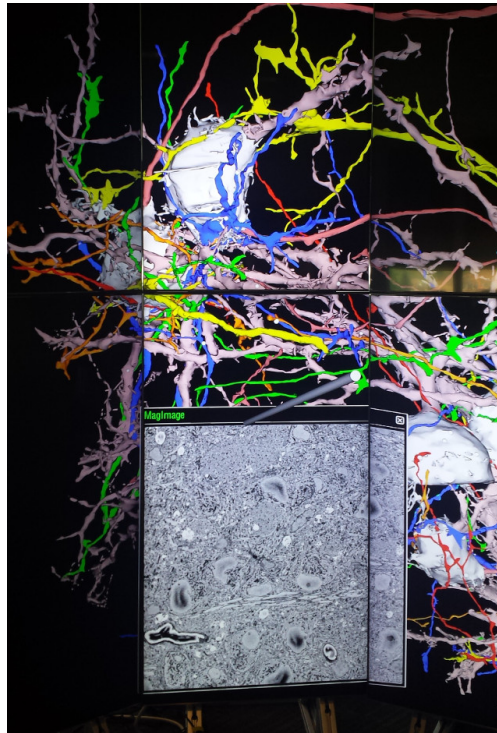


Figure 4.11: **Electron Microscopy Image Projection.**

4.2 Design and Implementation: Unity3D

4.2.1 Motivation

Despite the many advantages that the CAVE design offers, there are several practical disadvantages. The upfront investment cost makes the systems hard to obtain for many scientific labs. Because of the design of CalVR and its reliance on many open source dependencies, cross platform support is difficult to obtain, and installation on various Linux operating systems tedious and error prone. Due to the physical size of CAVE systems, the portability is strongly restricted, which reduces users ability to collaborate and share. These three issues lower the usability of BrainTrek on a grander scale.

4.2.2 Unity and BrainTrek

To allow the use of BrainTrek to everyone, the team decided to port the functional aspects of the system to the Unity3D engine. There are many innate advantages of Unity: built-in graphical

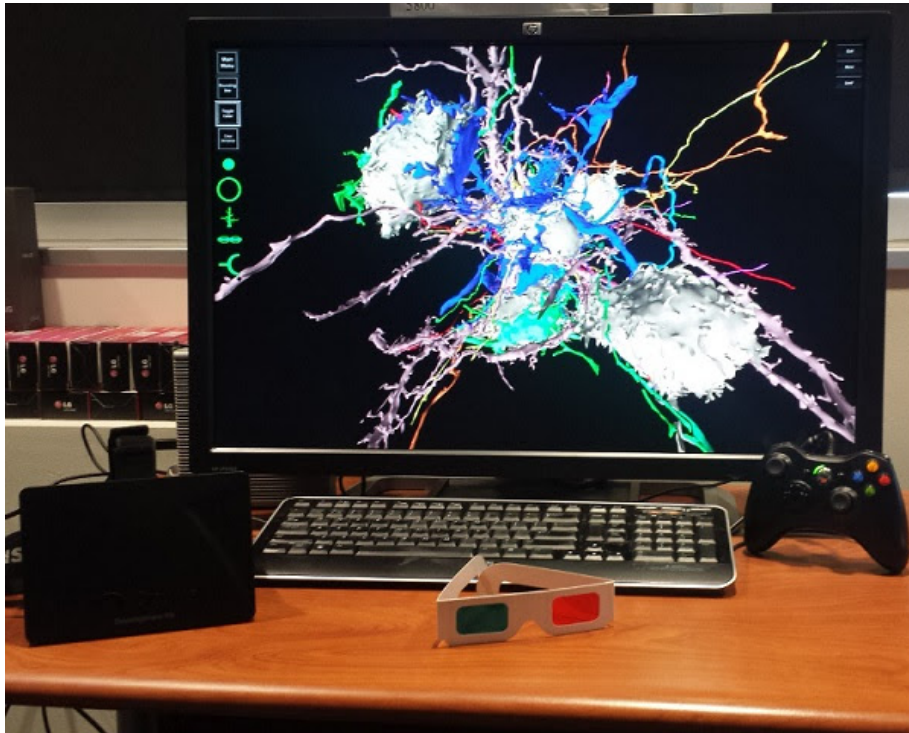


Figure 4.12: **BrainTrek on Unity.**

optimizations, an easy-to-use GUI for set design, and comparatively fast development time. But Unity also provides certain features that negate the CAVE systems short comings. Single-Screen BrainTrek (ssBrainTrek) allows users on any platform (Windows, OS X, Linux) to visualize the CAVE environment directly on their computer screen. It is also compatible with Unitys web player framework, so users may simply load a scene from a web browser, providing further access to the system.

Regular computer monitors lack depth; visualizing 3D scenes are not comparable to the CAVE, so ssBrainTrek supports four viewing modalities. The conventional 2D computer screen is available for regular computers. Users with anaglyph glasses (red-cyan) can get a sense of simulated depth with the Anaglyph modality. This provides depth at the lowest possible cost. Users with head-mounted displays (HMDs), like the Oculus Rift, can seamlessly switch into the HMD modality for a true 3D experience. With a base cost of 300 USD, the Oculus Rift provides a low cost solution for scientific labs attempting to view 3D models. Finally, 3d TVs are now supported as a fourth viewing modality.

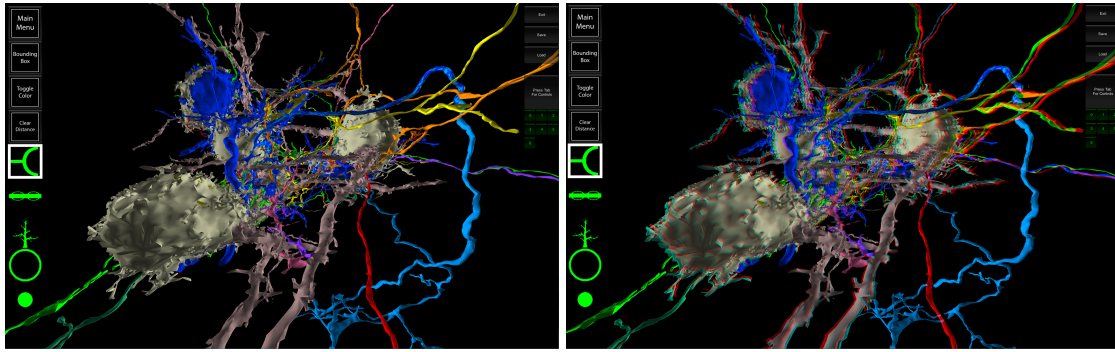


Figure 4.13: **BrainTrek, Regular View and Anaglyphic Mode.**

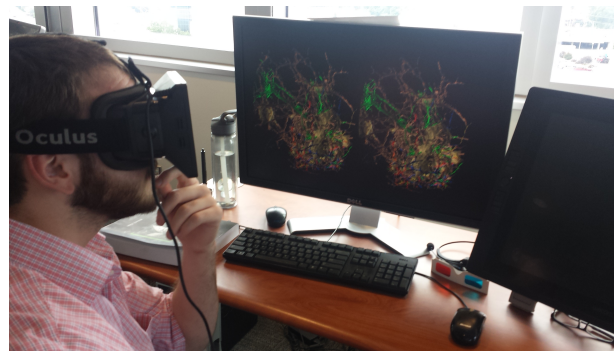


Figure 4.14: **BrainTrek Viewed With Oculus Rift.**

4.2.3 ssBrainTrek

ssBrainTrek is the visualization/exploration system for BrainTrek on Unity. This is akin to the CAVE version of BrainTrek — users can fly around cells, exploring different angles and positions, as well as modifying the transparency of cellular parts for ideal viewing. It supports scene hopping, so users can select different cellular ages to explore. Users may visualize in the four viewing modalities (conventional, anaglyph, 3D TV, and HMD). While utilizing a HMD, users are unable to locate the keyboard and mouse, so Cell Explorer also supports the use of game controllers (Xbox USB controller).

Some analytical tools have been built into Cell Explorer. Distance measurements are easily determined by right-clicking on two difference points. After each click, a small sphere is generated on the cell model. Once two spheres are located in the scene, a thick white line is drawn between the spheres, and the distance is reported in the corner. This line and distance measurement can be

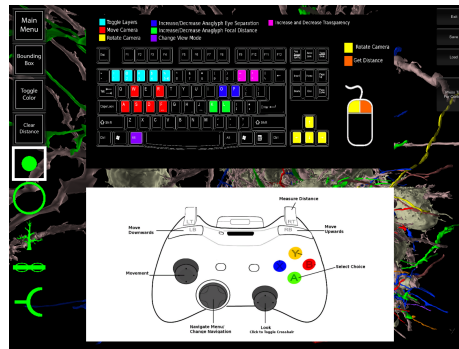


Figure 4.15: **User Controls for Both Systems.**

cleared for un-obscured viewing. Previously mined statistics on model volumes, surface areas, and synapse surface areas can be reported on each cellular part. These two tools allow users to quickly assess various information that might be important to their exploration (how long is this dendrite?). Finally, users can display the bounding box of the volume.

Sometimes while exploring the scene, it is useful for scientists to take notes or review previously examined scenes. To facilitate this, a Position Saving feature has been created. This feature captures the current view, the user's name and description, and allows for an audio note to be recorded. After entering their pertinent information, users may record an audio note of up to five minutes in length. When returning to the scene via the Load button, users may play back their note. This allows seamless dictation without having to look away from the scene or remove a HMD.

Controls

ssBrainTrek supports both keyboard and mouse support, as well as standard game controllers. We utilized a low-cost USB-connected XBOX One controller for testing. In preliminary usage, HMD users preferred the game controller system as it was easier to locate while wearing an Oculus Rift.

4.2.4 Annotation Tools

Although exploration of neuronal volumes is the main aspect of the system, the BrainTrek system can be modified into sub modules that are useful for cellular annotation. These sub modules support the three viewing modalities. Both report user findings to our own server via socket. The

server then uses a custom python script to save each session into a Mongo database.

Synapse Tagger

One of the main challenges in neuroscience is the creation of a connection map between neurons, or a connectome (White et al. 1986). For a connectome to be created, the location and size of synapses must be detected in a quick and efficient way. Previously, synapse connections have been hand-located and traced, allowing the quantification of the surface area of the synapse; however the process is time-intensive and prone to subjective judgments by the tracer. To achieve faster synapse detection and modeling, weve developed a semi-automated process entitled Synapse Tagger.

SynTag works in a straight-forward method. The user is presented with a single cell, including its soma, dendrites, axon, and neuronal inputs from other cells. The user may fly around the cell and visually search for locations where neuronal inputs become very close to the soma or dendrite of the cell. These close proximity locations are characteristic of synaptic contact. The user then aims a ray-cast via mouse or on-screen aiming reticle at the location. The ray-cast is fired, retrieving the exact coordinates of the synapse on the side of the neuronal input. The user may then toggle the visualization of the input off, allowing a clear view of the soma or dendrite. The user fires a second ray-cast, which captures the coordinates of the synapse on the soma or dendrite side. The midpoint between these coordinates is calculated. This midpoint is the assumed location of a synapse. SynTag collects information of the input model name, the soma/dendrite model, the location of the synapse, and the relative size of the synapse surface. The information is then saved to a CSV file. SynTag allows for potential synapses to be tagged quickly, approximately 15 synapses per minute.

Once the synapses have been tagged by humans in SynTag, the tagged locations are then used to filter the automatic detection of synaptic surface area. The algorithm restricts its search to regions identified by the user in SynTag, and then identifies points on the input within that region that are very close to the soma/dendrite of the cell. Once a list of points has been identified, the automatic surface area generator composes a new mesh of the member faces on the input that those points are a member of, and uses this new mesh to calculate the precise surface area of the synapse between the input and the cell.

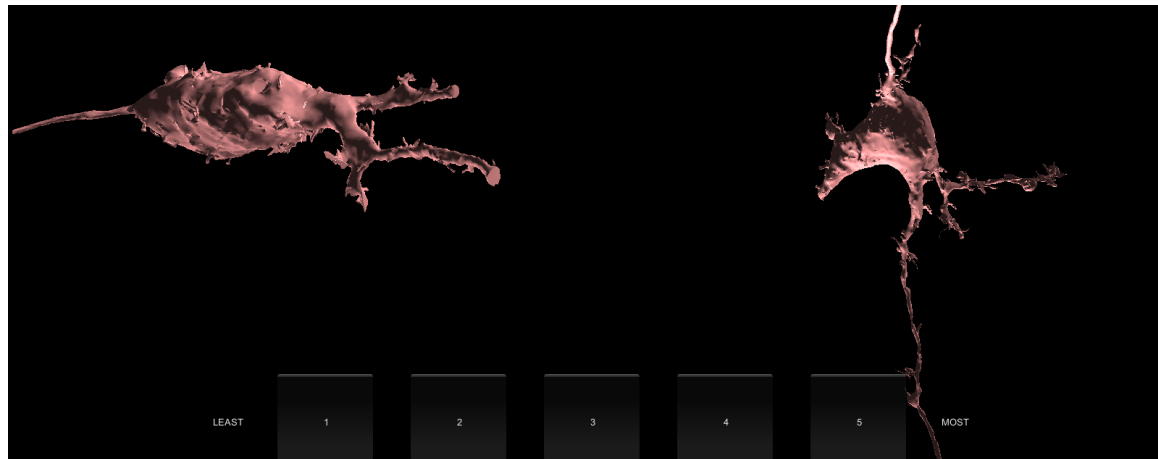


Figure 4.16: **Cell Comparison.** Two cells are compared. Morphological similarity is rated on a 1 to 5 scale.

Cell Comparison

Establishing a ground truth of cell identity is an important step in training a machine to classify neurons. Cell Comparison (CellComp) allows users to rate cells based on their similarity to each other, which provides a human-created ground truth. Upon loading, a base cell is randomly selected out of the group. Users are presented with the base cell and an alternate cell. After visually comparing the two, users rate the alternate cell based on its perceived similarity to the base cell. The cells rating is then sent to a server, where the results are stored for future analysis.

Chapter 5

Experimental Results

5.1 Experiments

This section evaluates several aspects of BrainTrek. Finite rendering power forces to consider a balance between model accuracy and polygon reduction. Section 5.1.1 examines the lower resolution limits at which relevant neuronal features become visible. Section 5.1.2 tests the rendering frame rate against the model size expressed in number of model vertices. Section 5.1.3 computes the rendering frame rate against the size of a cell set, and Section 5.1.4 describes a user evaluation of the user interface.

5.1.1 Neuron Resolution

To our knowledge, a qualitative examination of the necessary model resolution for neural processes has not been undertaken. Since finite rendering power limits visualization of large volumes, a balance point between model accuracy and GPU-load should be determined. We identify a small list of neuronal features and rate their visibility at various model resolution. Figure 4.6A shows the full resolution model at 64 vertices/ μm^2 . At 8.2 vertices/ μm^2 , fine processes called neurites (white arrow) are visible (see Figure 4.6B). Below 5 vertices/ μm^2 , fine macro processes are incorrectly shortened due to mesh decimation approximations. At this point, the tips of the dendrites and axons reach their approximate location (see Figure 4.6C). With any resolution lower than 2.3 vertices/ μm^2 , the neurons appear blocky. It is immediately obvious that such a representation could

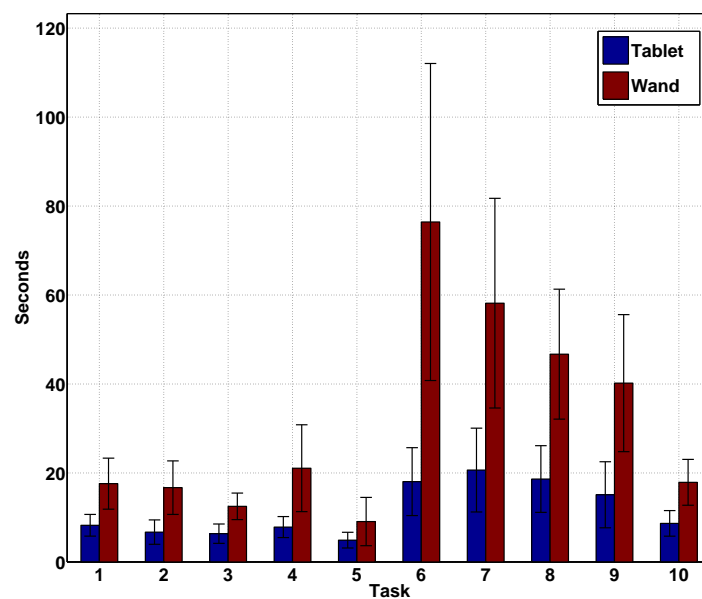


Figure 5.2: **Tablet vs Wand.** User interaction times are measured for specific tasks for both the wand and tablet. The first five tasks only involve global changes to the set of cells, while the latter five involve specific cell selection and modification.

not be accurate (see Figure 4.6D). Figure 4.5, left, summarizes the resolution intervals at which neuronal features are visible.

5.1.2 Frame Rate

In order to quantify system performance and efficiency, we ran a series of tests measuring the frames per second (fps) compared to the total amount of vertices in a particular scene. The more vertices rendered, the slower the frames per second. The system has a ceiling of 60 fps. Below 10-15 fps it becomes significantly less responsive. Above 30 fps, performance gains are practically unnoticeable.

By comparing the amount of vertices to frame rates, we made a few key observations. Beyond 300,000 vertices, frame rates began to dip <30fps. At about 600,000 vertices, responsiveness suffers as frame rates descend to about 15fps. Near 2,000,000 vertices, we experienced frame rates of 5fps or lower, yielding a particularly unwieldy sluggish viewing experience (see Figure 4.5, center).

TASK LIST		SURVEY		
Section 1: Part Modification		Quality	Tablet	Wand
1	Set the opacity of inputs to 50%	Intuitive	10	4
2	Set the opacity of dendrites to 75%	Physical Comfort	14	0
3	Set the opacity of nuclei to 25%	Overall Preference	12	2
4	Set all parts invisible (0% opacity)			
5	Fully display all parts (100% opacity)			
Section 2: Cell Selection and Manipulation				
6	Make only cell 8 and 9 invisible			
7	Set the opacity of cell 4's nucleus to 50%			
8	For all cells except 1, 2, and 3, set the opacity of the axons to 0%			
9	Return all parts of all cells except 8 and 9 to 100% opacity			
10	Return all parts of all cells to 100% opacity			

Table 5.1: **Task list and comparative survey.** Left: Each participant was educated on a specific device and asked to perform the tasks in this list. Right: Once participants had tried both input devices, they selected which device they preferred for various qualities.

5.1.3 Cell Set Size

The number of vertices per cell varied greatly. However, on average, a neuron has 264,207 vertices. Therefore, the display of only five cells was enough to lower the frame rate down to 12fps. At nine cells, the frame rate hovered at around 5fps. To prevent this, mesh decimation is utilized to reduce the vertices per cell, allowing for more cells per scene and higher frame rates.

Because mesh decimation can reduce a model to a percentage of its total vertices, a lower limit of cell presentation can be easily calculated. With the average cell surface area and average vertices, we compute our full resolution models at approximately $64 \text{ vertices}/\mu\text{m}^2$. With the average reduced vertex count, we calculate our minimum resolution at $8.23 \text{ vertices}/\mu\text{m}^2$. On average, our cell models possess a surface area of $6373.80\mu\text{m}^2$. Therefore, we can normalize the system frame rate experiment to predict the frame rate at the lowest resolution (see Figure 4.5, right).

5.1.4 UI Evaluation

Studies on CAVE interaction have shown strengths and weaknesses in all input devices. Quantitative analysis of usability metrics like time per task have been studied, and qualitative surveys have assessed user feedback factors like ease of use. We performed a study on the comparative advantages of our wand interaction system and the tablet interface. 14 participants were recruited

to participate by word of mouth. Each was randomly assigned to either Group A or Group B. The groups only dictate which input device the participant tested first. A participant would enter the lab and be introduced to their first system. A five minute tutorial was provided on the system and the input device. Participants were then asked to perform a series of tasks manipulating a nine cell data set projected in 3D (see Table 5.1, left, for the list of tasks). These tasks could be attempted until the user gave up. Task completion and time measurements were recorded (see Figure 5.2 for the average time comparison). Once the series was complete, the participant was given a three question survey on their experience. The input device was switched to the alternative and the process repeated. Finally, the participants were given a survey comparing the two.

Overall, users performed faster via tablet interaction. Although tasks related to part transparency ranked somewhat similarly, tasks related to cell selection were greatly enhanced by the touch screen. Participants often exclaimed frustration with the wand, as precise adjustments to the sliders is more difficult. Qualitatively, participants found the tablet easier to learn and use, as well as more enjoyable overall (see Table 5.1, right, for a response summary to some key questions). Some participants found the wand to be more intuitive. A few verbally expressed their enjoyment of the “futuristic” nature of the wand. In terms of outright performance, however, the tablet proves a better user interface.

Chapter 6

Conclusion

BrainTrek has proven to be an intuitive UI that is capable of manipulating the depiction of sufficient numbers of cells to provide a useful and pleasing IVR experience. A novel interface with pre-processing has been designed to adjust model complexity that works well with a middle price point hardware system that can be accessible for many laboratories. Benchmarks have been established to scale hardware capabilities upward (and cost) for display of larger numbers of models using the same number of screens or a larger screen system for viewing by more than three people.

The system has been expanded via the Unity engine. This provides a more accessible application for a larger audience of users. The Unity based BrainTrek is easy download and install. It supports two control systems and four viewing modalities. The platform is quick to develop, allowing additional plug-ins to be developed rapidly. Currently, two annotation plug-ins have been developed, one developed to cluster similar morphological structures, and one developed to located and annotate potential synaptic regions.

Both of these tools can be later used in future work, where the implementation of machine learning techniques will be applied for automated synaptic detection and morphological classification. Further development of analytical tools is planned. Online collaboration should be supported for 'multiplayer' exploration of cell volumes. Although BrainTrek is designed for neuron models, other scientific models, like proteins or DNA, could easily be used, opening the scope to a general scientific 3D viewing system. There is much to be developed and discovered on our continued trek through the brain.

References

- [1] Daniel Gardner, Huda Akil, Giorgio A Ascoli, Douglas M Bowden, William Bug, Duncan E Donohue, David H Goldberg, Bernice Grafstein, Jeffrey S Grethe, Amarnath Gupta, et al., “The neuroscience information framework: a data and knowledge environment for neuroscience,” *Neuroinformatics*, vol. 6, no. 3, pp. 149–160, 2008.
- [2] Paarth Chothani, Vivek Mehta, and Armen Stepanyants, “Automated tracing of neurites from light microscopy stacks of images,” *Neuroinformatics*, vol. 9, no. 2-3, pp. 263–278, 2011.
- [3] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner, “The structure of the nervous system of the nematode *caenorhabditis elegans*,” *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, vol. 314, no. 1165, pp. 1–340, 1986.
- [4] Kevin L Briggman and Davi D Bock, “Volume electron microscopy for neuronal circuit reconstruction,” *Current opinion in neurobiology*, vol. 22, no. 1, pp. 154–161, 2012.
- [5] Davi D. Bock, Wei-Chung A. Lee, Aaron M. Kerlin, Mark L. Andermann, Greg Hood, Arthur W. Wetzel, Sergey Yurgenson, Edward R. Soucy, Hyon S. Kim, and R. Clay Reid, “Network anatomy and in vivo physiology of visual cortical neurons,” *Nature*, vol. 471, no. 7337, pp. 177–182, Mar. 2011.
- [6] Kevin L. Briggman, Moritz Helmstaedter, and Winfried Denk, “Wiring specificity in the direction-selectivity circuit of the retina,” *Nature*, vol. 471, no. 7337, pp. 183–188, Mar. 2011.
- [7] P. Holcomb, T. Deerinck, M. Ellisman, and G. Spirou, “Construction of a polarized neuron,” *Journal of Physiology*, vol. 591, pp. 3145–50, 2013.
- [8] Moritz Helmstaedter, “Connectomics: The dense reconstruction of neuronal circuits,” *Frontiers in Neuroinformatics*, , no. 16.
- [9] Zhiping Lai, Griselda Bonilla, Isabel Diaz, Jose Geraldo Nery, Khristina Sujaoti, Miguel A Amat, Efrosini Kokkoli, Osamu Terasaki, Robert W Thompson, Michael Tsapatsis, et al., “Microstructural optimization of a zeolite membrane for organic vapor separation,” *Science*, vol. 300, no. 5618, pp. 456–460, 2003.
- [10] Winfried Denk and Heinz Horstmann, “Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure,” *PLoS Biol*, vol. 2, no. 11, pp. e329, 10 2004.

- [11] “Eyewire,” June 2012, <http://eyewire.com/>.
- [12] Bo Huang, Bin Jiang, and Hui Li, “An integration of gis, virtual reality and the internet for visualization, analysis and exploration of spatial data,” *International Journal of Geographical Information Science*, vol. 15, no. 5, pp. 439–456, 2001.
- [13] Christoph H Arns, Mark A Knackstedt, W Val Pinczewski, and Nicos S Martys, “Virtual permeametry on microtomographic images,” *Journal of Petroleum Science and Engineering*, vol. 45, no. 1, pp. 41–46, 2004.
- [14] Robert Kosara, Christopher G Healey, Victoria Interrante, David H Laidlaw, and Colin Ware, “Thoughts on user studies: Why, how, and when,” *IEEE Computer Graphics and Applications*, vol. 23, no. 4, pp. 20–25, 2003.
- [15] Xuwei Liang, Ning Kang, Stephen E. Rose, Jonathan B. Chalk, Jun Zhang, and Dr. Jun Zhang, “Diffusion tensor imaging analysis of regional whiter matter changes along the cingulum in mild cognitive impairment,” .
- [16] Andries Van Dam, David H Laidlaw, and Rosemary Michelle Simpson, “Experiments in immersive virtual reality for scientific visualization,” *Computers & Graphics*, vol. 26, no. 4, pp. 535–555, 2002.
- [17] Thomas A DeFanti, Daniel Acevedo, Richard A Ainsworth, Maxine D Brown, Steven Cutchin, Gregory Dawe, Kai-Uwe Doerr, Andrew Johnson, Chris Knox, Robert Kooima, et al., “The future of the cave,” *Central European Journal of Engineering*, vol. 1, no. 1, pp. 16–37, 2011.
- [18] Brad Myers, Scott E Hudson, and Randy Pausch, “Past, present, and future of user interface software tools,” *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 7, no. 1, pp. 3–28, 2000.
- [19] Tamer Demiralp, Zubeyir Bayraktaroglu, Daniel Lenz, Stefanie Junge, Niko A Busch, Burkhard Maess, Mehmet Ergen, and Christoph S Herrmann, “Gamma amplitudes are coupled to theta phase in human eeg during visual perception,” *International Journal of Psychophysiology*, vol. 64, no. 1, pp. 24–30, 2007.
- [20] Jürgen Schulze, Andrew Forsberg, Alexander Kleppe, Robert Zeleznik, and David H Laidlaw, “Characterizing the effect of level of immersion on a 3d marking task,” in *proceedings of HCI International*, 2005, pp. 447–452.
- [21] CHARLES WHEATSTONE, “Contributions to the physiology of vision. part the first. on some remarkable, and hitherto unobserved, phenomena of binocular vision.,” *Philosophical Transactions” of the Royal Society of London*, Vol. 128, pp. 371 - 394., 1838.
- [22] Carolina Cruz-Neira, Daniel J Sandin, and Thomas A DeFanti, “Surround-screen projection-based virtual reality: the design and implementation of the cave,” in *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. ACM, 1993, pp. 135–142.

- [23] Alessandro Febretti, Arthur Nishimoto, Terrance Thigpen, Jonas Talandis, Lance Long, JD Pirtle, Tom Peterka, Alan Verlo, Maxine Brown, Dana Plepys, et al., “Cave2: a hybrid reality environment for immersive simulation and information analysis,” in *IS&T/SPIE Electronic Imaging*, 2013, pp. 864903–864903.
- [24] Andrew Forsberg, Michael Katzourin, Kristi Wharton, Mel Slater, et al., “A comparative study of desktop, fishtank, and cave systems for the exploration of volume rendered confocal data sets,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 3, pp. 551–563, 2008.
- [25] Gil Bohrer, Marcos Longo, David J Zielinski, and Rachael Brady, “Vr visualisation as an interdisciplinary collaborative data exploration tool for large eddy simulations of biosphere-atmosphere interactions,” in *Advances in Visual Computing*, pp. 856–866. 2008.
- [26] Rachael Brady, John Pixton, George Baxter, Patrick Moran, Clinton S Potter, Bridget Carragher, and Andrew Belmont, “Crumbs: a virtual environment tracking tool for biological imaging,” in *Biomedical Visualization, 1995. Proceedings. IEEE*, 1995, pp. 18–25.
- [27] Dirk Rantzau, Karin Frank, Ulrich Lang, Daniela Rainer, and Uwe Wössner, “Covise in the cube: an environment for analyzing large and complex simulation data,” in *Proceedings of the 2nd Workshop on Immersive Projection Technology*, 1998, vol. 2001.
- [28] Henrik Tramberend, “Avocado: A distributed virtual reality framework,” in *Virtual Reality, 1999. Proceedings., IEEE*. IEEE, 1999, pp. 14–21.
- [29] J. P. Schulze, A. Prudhomme, P. Weber, and T. A. DeFanti, “Calvr: an advanced open source virtual reality software framework,” in *SPIE The Engineering Reality of Virtual Reality*, 2013.
- [30] Nguyen Thong Dang, Monica Tavanti, Ivan Rankin, and Matthew Cooper, “A comparison of different input devices for a 3d environment,” *International Journal of Industrial Ergonomics*, vol. 39, no. 3, pp. 554 – 563, 2009.
- [31] Doug A. Bowman and Chadwick A. Wingrave, “Design and evaluation of menu systems for immersive virtual environments,” in *Virtual Reality, 2001. Proceedings. IEEE*, 2001, pp. 149 – 156.
- [32] Robert W. Lindeman, John L. Sibert, and James K. Hahn, “Hand-held windows: Towards effective 2d interaction in immersive virtual environments,” in *In IEEE Virtual Reality*, 1999, pp. 205–212.
- [33] Chreston A. Miller, Ashley Robinson, Rongrong Wang, Pak Chung, and Francis K. H. Quek, “Interaction techniques for the analysis of complex data on high-resolution displays,” in *ICMI*. 2008, pp. 21–28, ACM.
- [34] Neil G Smith, Steve Cutchin, Richard A Ainsworth, Daniel J Sandin, Jurgen Schulze, Andrew Prudhomme, Falko Kuester, Thomas E Levy, Thomas A DeFanti, and Robert Kooima, “Cultural heritage omni-stereo panoramas for immersive cultural analytics—from the Nile to the hijaz,” .

- [35] “Zygote body browser,” June 2010, <http://www.zygotebody.com/>.
- [36] K Siau, “Human-computer interaction: The effect of application domain knowledge on icon visualization,” *Journal of Computer Information Systems*, vol. 45, no. 3, pp. 53–62, 2005.
- [37] J. Kremer, “Computer visualization of three-dimensional image data using imod,” *Journal of Structural Biology*, vol. 116, no. 1, pp. 71–76, Jan. 1996.
- [38] William J Schroeder, Jonathan A Zarge, and William E Lorensen, “Decimation of triangle meshes,” in *ACM SIGGRAPH Computer Graphics*. ACM, 1992, vol. 26, pp. 65–70.