A Topological Framework for Semi-Automoatic Neuron Tracing in Virtual Reality

Torin McDonald
University of Utah

UUCS-19-004

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

30 April 2019

Abstract

Researchers in the field of connectomics are working to reconstruct a map of neural connections in the brain, in order to understand at a fundamental level how the brain processes information. Constructing this wiring diagram is done by tracing neurons through high resolution image stacks acquired with fluorescence microscopy imaging techniques. While a large number of automatic tracing algorithms have been proposed, these frequently rely on local features in the data and fail on noisy data or ambiguous cases, requiring time consuming manual correction. As a result, manual and semi-automatic tracing methods remain the state-of-the-art for creating accurate neuron reconstructions. We propose a new semi-automatic method which uses topological features to guide users in tracing neurons and integrate this method within a virtual reality (VR) framework previously used for manual tracing. Through evaluation with experts we find that our topologically guided approach is able to accurately trace neurons and improves trace time compared to both manual tracing in VR and existing semi-automatic tracing methods. Furthermore, users reported the topology guided tool to be less fatiguing, and more helpful when resolving noisy or low resolution regions.

A TOPOLOGICAL FRAMEWORK FOR SEMI-AUTOMATIC NEURON TRACING IN VIRTUAL REALITY

by

Torin McDonald

A Senior Thesis Submitted to the Faculty of
The University of Utah
In Partial Fulfillment of the Requirements for the Degree

Bachelor of Computer Science

School of Computing
The University of Utah
May 2019

Approved:

_____	_____
Valerio Pascucci, PhD Thesis Faculty Supervisor	H. James de St. Germain, PhD Director of Undergraduate Studies School of Computing

_____
Ross Whitaker, PhD
Director
School of Computing

ABSTRACT

Researchers in the field of connectomics are working to reconstruct a map of neural connections in the brain, in order to understand at a fundamental level how the brain processes information. Constructing this wiring diagram is done by tracing neurons through high resolution image stacks acquired with fluorescence microscopy imaging techniques. While a large number of automatic tracing algorithms have been proposed, these frequently rely on local features in the data and fail on noisy data or ambiguous cases, requiring time consuming manual correction. As a result, manual and semi-automatic tracing methods remain the state-of-the-art for creating accurate neuron reconstructions. We propose a new semi-automatic method which uses topological features to guide users in tracing neurons and integrate this method within a virtual reality (VR) framework previously used for manual tracing. Through evaluation with experts we find that our topologically guided approach is able to accurately trace neurons and improves trace time compared to both manual tracing in VR and existing semi-automatic tracing methods. Furthermore, users reported the topology guided tool to be less fatiguing, and more helpful when resolving noisy or low resolution regions.

CONTENTS

ABSTRACT . . . . . **i**

1. INTRODUCTION **1**

2. TECHNICAL BACKGROUND **3**

2.1 Morse Functions 3
2.2 The Morse-Smale Complex 4
2.2.1 Persistence and Cancellation 5
2.2.2 Algorithmic Approaches 6
2.3 Applications of the MS Complex 8

3. RELATED WORK **9**

3.1 Neuron Tracing Workflow 9
3.2 Automatic and Semi-Automatic Neuron Tracing 10
3.3 Immersive Environments 11

4. TOPOLOGY GUIDED NEURON TRACING **12**

4.1 Computing Ridge Graphs 12
4.2 Virtual Reality Neuron Tracing Framework 14
4.2.1 Tracing and Navigation 15
4.2.2 Rendering 16
4.3 Semi-Automatic Tracing in Virtual Reality 17
4.3.1 Tracing 18
4.3.2 A Fast and Efficient Querying Framework 19

5. EVALUATION **21**

5.1 Offline Comparison 22
5.2 Expert Evaluation 24
5.3 Discussion 27

6. SUMMARY AND FUTURE WORK **32**

REFERENCES **34**

INTRODUCTION

A central goal within the field of neuroscience is to understand how the dense interconnected neural circuits in the brain communicate and process information, and how this processing relates to behavior. The field of connectomics was founded to understand the fundamental wiring map of the brain, to comprehend these neural circuits at a mechanistic level. Through analyzing neuron structure and connectivity, neuroanatomists can gain a deeper understanding of fundamental brain functions, leading to new insights about brain diseases and potential treatments.

However, obtaining a comprehensive wiring diagram for even relatively small and simple mammalian brains, such as that of a mouse, is a massive undertaking [5, 9, 44, 54]. Projects focusing on species with larger brains more similar to humans, such as non-human primates (NHP) are even more challenging. Although recent advancements in high-resolution tissue labeling [36], optical tissue clearing [11, 42, 58] and imaging [44, 54] have made it possible to image NHP brains at large scales and high-resolutions, the technology for extracting the imaged neuron morphologies has struggled to keep up.

Current efforts to improve the speed of neuron morphology extraction have largely focused on fully automatic techniques. Automatic techniques take a stack of images and attempt to extract the imaged neuron structures, without user input. The DIADEM (DIgital reconstructions of Axonal and DEndritic Morphology) Challenge [21] was proposed in 2009 to motivate improvement of these techniques. The ultimate goal of this community effort was to increase the speed that neurons could be traced by $20\times$. However, at the end of the challenge no algorithm had achieved this goal due to the laborious post-processing required to correct errors [35]. Peng et al. [48] reported that this post-processing step can take longer than a manual tracing. Although additional efforts to improve automatic reconstruction are ongoing [49], in practice the bulk of neuron tracing is done manually [39] or with a semi-automatic method.

Manually tracing neurons is a difficult and time consuming process. Tracing is typically done on a desktop, using standard software such as NeuroLucida [38] or Vaa3D [50]. The data is displayed as a 2D set of images or 3D volume, and the user clicks along the neuron

to draw a line. The lack of ability to directly make selections in 3D or navigate the data in 3D introduces additional usability challenges on top of the already difficult task of tracing. To address this issue, Usher et al. [56] proposed a Virtual Reality (VR) based tool for manual neuron tracing and found that neuroscientists using the tool performed similar quality traces in less time.

Semi-automatic neuron tracing methods have been proposed to provide a compelling alternative to both manual and fully automatic neuron tracing [37,43,47,50]. When using a semi-automatic method the user provides coarse guidance to the algorithm, e.g., through a set of start and end points or clicks. The algorithm then extracts the neuron structure between these guide points. Semi-automatic methods can significantly reduce the amount of time taken to trace a neuron by integrating the neuroscientist's guidance into the algorithm, reducing the amount of post-processing manual cleanup required.

In this work we propose a new semi-automatic neuron tracing framework which builds off topological analysis methods [22]. Our approach uses the Morse-Smale Complex to precompute a superset of potential paths that follow neurons. Having access to this superset of traces allows neuroscientists to quickly trace along the neuron of interest by selecting subsets of these paths. We implement our semi-automatic method within a virtual reality neuron tracing system to provide an intuitive environment to work with the 3D data. In a pilot study with neuroanatomists we find that our approach provides significant benefits, retaining trace accuracy while improving speed and reducing fatigue. Our contributions are:

- A novel topologically guided framework for real-time semi-automatic neuron tracing;
- An intuitive interaction design for using this framework in VR; and
- A comparison of our approach against widely used semi-automatic methods as well as previous manual tracing methods.

TECHNICAL BACKGROUND

Computational topology is a field that combines mathematics and computer science to determine topological features through the efficient use of algorithms. In recent years computational topology has played an important role in visualizing and analyzing scientific data, especially for large and complex data sets. Topology allows for the analysis of how features in data are connected, rather than focusing on raw spatial relationships in data. Important structural features in scientific data sets can be extracted to enhance visualization and analysis and remove extraneous noise.

2.1 Morse Functions

Morse Theory, a fundamental building block of differential topology, is useful for analyzing the topological features in context of real valued smooth functions. While every derivative of a function must exist in order for it to be smooth, for the purpose of Morse theory studied in a computational topology context, only the first and second derivatives of the function must be defined [16].

Let M be a smooth d-manifold. Consider a real valued smooth mapping $f : M \to R$. This mapping induces a linear mapping between the tangent spaces TM and TR, which is the derivative Df. That is to say, the derivatives of f are real valued linear maps on the tangent spaces. A point x that exists on manifold M is a *critical point* if the first order partial derivatives of points in a neighborhood of x are equal to zero. In other words, for x to be critical, the gradient of f at x must be zero. To further classify the critical points, the second order derivatives are used. A critical point is *non-degenerate* if the Hessian of f, the matrix of its second derivatives is non singular, or the determinant of the Hessian matrix is not equal to zero. The function $f : M \to R$ is a *Morse function* if all of the critical points on M are non-degenerate, and if the critical points have distinct function values.

Morse Lemma: Let p be a non-degenerate critical point for f. Then there is a local coordinate system $(x_1 x_2, ..., x_n)$ in a neighborhood U of p with $x_n(p) = 0$ and such that the identity

$$f(x) = f(p) - x_1^2 - ... - x_q^2 + x_{q+1}^2 + ... + x_n^2 \tag{2.1}$$

holds throughout U [41].

The Morse Lemma provides a few key properties for Morse functions. First of all, it allows for the distinct classification of non degenerate critical points, which is the number of minus signs in the quadratic polynomial, and is independent by the index of the coordinates. This index also corresponds to the number of negative eigenvalues in the Hessian matrix. For a 2-dimensional manifold, the critical points can be classified based on these indices. A minimum has index 0, a saddle has index 1, and a maximum has index 2. The critical points in a 3 dimensional manifold are characterized by four possible index values instead of three: a minimum of index 0, a 1-saddle of index 1, a 2-saddle of index 2, and a maximum of index 3. In addition to lending itself to a classification of critical points, the Morse Lemma also indicates that non-degenerate critical points are isolated, and therefore there are a finite number of critical points on a compact manifold.

2.2 The Morse-Smale Complex

Given a Morse function f, the manifold is decomposed by following the gradient. To do this, we define an *integral line* that follows this gradient field starting at a critical point and ending at another, the critical points being the *origin* and *destination* respectively. We can now decompose f into manifolds of its critical points and classify these manifolds as either an *descending manifold* and a *ascending manifold*. The *descending manifold* of a critical point u is the point itself together with all regular points whose integral lines end at u. The *ascending manifold* of a critical point u is the point itself together with all regular points whose integral lines originate at u [16]. The Morse function f is a *Morse-Smale function*

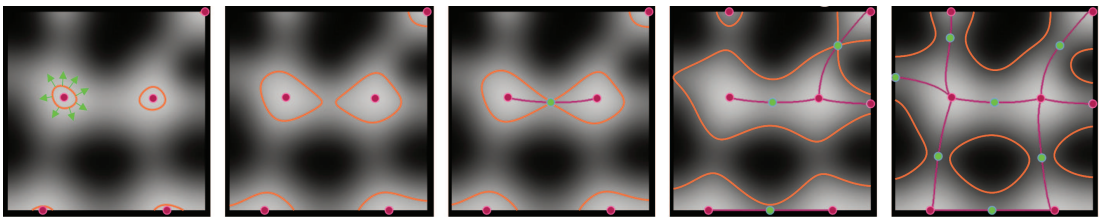


Figure 2.1: The MS complex tracks the topology of level sets in a sweep from ∞ to $-\infty$ (left to right). New level set components are created at maxima (red dots), and either join or split at saddles (green dots). The 1-skeleton of the MS complex is orthogonal to the level sets at all points. At each point in the sequence, we show the portion of the 1-skeleton above the level set. Note that the saddle-maximum lines (red) trace ridge-like structures.

if the ascending and descending manifolds of f intersect transversely. By extension, the *Morse-Smale complex* (MS complex) is the intersection of the ascending and descending manifolds of f.

The 0- and 1-dimensional cells of the intersection of ascending and descending manifolds form the *1-skeleton* of the Morse-Smale complex (Fig. 2.1). Practically, the MS complex *1-skeleton* is formed by *nodes* and *arcs*. *Nodes* are the critical points of the MS complex, and *arcs* are the integral lines connecting critical points which differ in index by one. For a complete visual overview of the components of the MS complex we refer to Gyulassy et al. [29].

2.2.1 Persistence and Cancellation

When using computational topology to analyze scientific data, noise within the data inherently adds topological features to the data that often distract from the desired features that are being extracted. The question arises: is there a way to remove topological features produced by noise in the data without changing the core topology being computed?

One solution, that has been shown to be particularly effective, is the notion of persistent homology. Persistence, at a high level, is a measure of how important a particular topological feature is in relation to the complete topology of the data. It is defined by the absolute difference in function value between a critical point pair. Topological features that are more likely to be attributed to noise have a low persistence, while features that are more likely to be important in the data set have a high persistence. Edelsbrunner et al. introduced the idea of topological persistence, and extended it as a method for simplifying a MS complex [18].

The MS complex can be simplified through the cancellation of critical point pairs under certain restrictions. A cancellation is valid if the indices of the critical points differ by one. This means that in two dimensions only saddle-extremum pair cancellations are allowed, and in 3D there arises the possibility of cancelling a 1-saddle and 2-saddle. As the Morse function f changes locally with every cancellation, the gradient vector field is smoothed, and therefore f is smoothed as well. Another feature that arises through the cancellation of critical points is the ability to create a structural hierarchy based on persistence. If cancellations are performed in order of persistence, then it is possible to have multiple

representations of the MS complex based on topological resolution.

2.2.2 Algorithmic Approaches

Because perfectly smooth functions are rarely found in scientific data, in order to compute the MS complex for non smooth functions it is necessary to either find a non smooth function that approximates a smooth function, or find a series of non smooth functions that approach the limit of a smooth function [16]. Two ways of accomplishing this have been studied, using either piecewise linear (PL) functions or utilizing discrete Morse theory.

An algorithm for computing the MS complex was first proposed by Edelsbrunner et. al. [15] The method was designed for computing the MS complex along a two dimensional PL function. Because PL functions are not smooth, and because line integrals along a PL function are ill-defined, the proposed algorithm set forward a number of steps to reach the final approximation of the MS complex. First, a quasi MS complex is computed on a given triangulation of the PL function through the creation of monotonic curves following a sequence of steepest edges in the triangulation that never cross and always have infinitesimal distance from each other. At each critical point, these curves are either duplicated or extended in order to reduce the number of arcs per saddle. Next, the quasi MS complex is transformed into the MS complex through a sequence of transformations that follow the path of steepest ascent along the edges of the triangulation more closely. Edelsbrunner et al. also introduced the concept of persistence to create a hierarchy of MS complexes through cancellation of minimum-saddle and saddle-maximum critical point pairs.

This method for computing the MS complex of a triangulated 2D PL field was improved upon by Bremer et al. [7]. Rather than using the edges of the triangulation of the data, their approach constructed the MS complex by following the steepest lines of ascent or descent rather than the triangulation. In addition, a new hierarchy was proposed for simplifying the MS complex. In addition to having ordered cancellations based on persistence, this hierarchy is built on the segmentation of the MS complex into 'diamonds,' each having a one maximum and a one minimum as vertices. This allows independent cancellations of critical point pairs, which in turn allows cancellations for view dependent simplifications of the MS complex. Finally, the authors also introduced a new way of

reconstructing the geometry of the MS complex after a cancellation has been made by faithfully following the topology and steepest gradient within a certain error bound.

An algorithm for the computation of the MS complex for three dimensional data was again first proposed by Edelsbrunner et al. [17]. The algorithm was essentially an extension of the 2D quasi MS complex algorithm proposed in [15]. While the authors proposed the idea of a MS complex hierarchy, they did not provide an implementation. Because of the addition of a third type of critical point, there is no longer just one type of cancellation, e.i. saddle-extremum. In three dimensions, there must also be a method for saddle-saddle cancellation. The first comprehensive method for computing a hierarchy for the MS complex on a 3D manifold was proposed by Gyulassy et al. [24]. Unlike saddle-extremum cancellations, where the cancellation of a critical point pair can be thought as a merging of three critical points as one extremum, saddle-saddle cancellations may create new cells in the MS complex. In order to preserve the separation of the two extrema separated by the saddle-saddle pair, it is necessary to fill in the space with new cells by re-routing the arcs extending from the extrema. Note that this operation still smooths the function, and a future cancellation of one of the new saddle-extremum pairs will remove all of the newly created cells.

All of the approximations of the MS complex using PL functions are limited in their practical applications for large data sets because of the complexity of each algorithm. This was overcome by Gyulassy et al. [23], in which an algorithm for computing the MS complex based on a discretized domain, rather than using PL functions as an approximation for a smooth function, was proposed. This algorithm is based on discrete Morse theory, which encompasses the ideas presented in smooth Morse theory but for a discrete setting. This algorithm was significantly more efficient than the previous fastest algorithm [31], although it sacrificed some speed for better memory usage.

While all of the algorithms mentioned above, with the exception of [17], have been implemented to visualize the MS complex for scalar data, they still only produce an approximation of the MS complex in the form described in [15]. The geometric inaccuracies produced by these algorithms are generally the result of the greedy local assignment of gradient vectors that tend to accumulate large global errors. To more accurately capture the correct geometry for the MS complex Gyulassy et al. proposed two algorithms that

follow the gradient field of the data more faithfully. The first algorithm is a randomized algorithm that follows the framework of [23], but replaces the local optimization with carefully designed random selection. An important feature of this algorithm is that the standard deviation of the mean converges to 0 as mesh resolution increases. The second algorithm is a deterministic variant of the first, and integrates the probabilities of the first to directly extract near optimal geometry and connectivity [25].

The advances made by [23] and [25] have enabled the opportunity for accurate computation of the MS complex using normal desktop computational power, and thus enables the viability for using the MS complex as a framework for our semi automatic tracing method. We will use a parallel computation variant of [23] to compute the MS complex for neuron data sets.

2.3 Applications of the MS Complex

The components of the MSC have been used in practice to extract features of interest in a range of application domains. For example, these components define features in the electron density field in the quantum theory of atoms in molecules: maxima occur at atom locations; 2-saddle-maximum arcs are covalent bonds; and descending 3-manifolds are atomic basins [4]. In other domains, specially selected subsets of the MSC can be used to extract features: descending 2-manifolds represent bubbles in mixing fluids [34]; 2-saddle-maximum arcs can be used to extract the core of a porous solid [27] or the filamentary structure of galaxies [55]; descending 2- and 1-manifolds identify lithium diffusion pathways [28]; and ascending 2-manifolds define burning regions in combustion simulations [8]. In each application, the features of interest were computed by identifying the appropriate topological abstraction, and then selecting a subset of the topological features which correspond to the quantities under study.

While our work is inspired by these approaches, the images generated from fluorescence microscopy pose a massive challenge for automated analysis. In addition to high-intensity noise, images of neurons have uneven staining, shadows, alignment artifacts, and other unexplained gaps in the signal which require manual intervention. This poses a challenge to topological methods which report what is in the scalar function itself, faithfully representing artifacts and noise along with the desired signal.

RELATED WORK

The neuron morphology reconstruction workflow has a number of components, one of the most time consuming being the physical tracing of neurons. To provide context for neuron tracing we describe the typical reconstruction workflow in practice (Sect. 3.1). We then review current automatic and semi-automatic neuron tracing methods and their limitations (Sect. 3.2).

3.1 Neuron Tracing Workflow

Modern methods for acquiring neuron microscopy data use viral vectors carrying genes for fluorescent proteins [36]. When injected into the tissue these vectors induce fluorescence within the structures to be imaged, labelling them at high resolution. The brain tissue is then rendered optically transparent using a clearing technique such as CLARITY [11], PACT [58], or SWITCH [42], and imaged in blocks with a confocal or two-photon microscope. These methods allow for imaging large blocks of tissue or entire brains, and can produce terabytes of high-resolution image stacks.

To reconstruct the labeled neurons from these image stacks, neuroanatomists use commercial tools like NeuroLucida [38], or open-source tools like Vaa3D [50]. These tools display the collected image stacks as either a set of 2D slices or as a 3D volume, where the user can trace manually by drawing lines along the structures of interest, or guide a semi-automatic algorithm along the structures to extract them. Once the desired neurons have been reconstructed they can be used in brain function simulations or overlaid on top of functional maps of the brain, to understand the connectivity between brain regions.

Although fully automatic algorithms are also supported by standard tools, they are less widely used in practice due to issues with image or labeling quality and ambiguity. It is common for a lab to employ several trained undergraduates responsible for the bulk of the neuron tracing work, with additional tracing done by graduate students and research scientists.

3.2 Automatic and Semi-Automatic Neuron Tracing

Today, neuron tracing remains a crucial bottleneck in the field of connectomics [39]. A large body of work has been devoted to developing new methods to accelerate this process, either through fully automatic algorithms or semi-automatic user-guided algorithms.

A significant ongoing effort in the community has sought to develop and evaluate fully automatic algorithms for neuron reconstruction. Two community efforts, the DIADEM Challenge [21] and the ongoing BigNeuron Project [46, 49], seek to provide a test bed for evaluating new reconstruction algorithms. Results from the DIADEM challenge suggest that the current state of the art automatic tracing algorithms are not suitable for widespread use in practice. This is attributed to the significant manual post-processing effort required to correct the output from the algorithm [48]. For a full review of recent advances in automatic neuron reconstruction we refer to the survey by Acciai et al. [3].

Due to the challenges in using fully automatic methods in practice, semi-automatic algorithms have found a growing interest in the community. When using a semi-automatic reconstruction algorithm, the user guides the algorithm along the neuron by tracing roughly along the neuron or clicking to mark start, branch, and end points to connect. By integrating more guidance from the neuroscientist into the algorithm the amount of additional post-processing cleanup required can be reduced, while still decreasing the time spent tracing compared to a fully manual trace. For example, Vaa3D's semi-automatic approach uses a pixel based shortest path algorithm [47] to connect the start point and one or more markers placed by the user. NeuroLucida 360's [37] semi-automatic tracing works similarly, where the user traces along the feature to guide the algorithm to important features. Neuromantic [43] uses a 3D extension of Meijering et al.'s 2D steerable Gaussian filter algorithm [40] for semi-automatic reconstruction.

However, these methods all work in the context of traditional desktop software, taking 2D inputs from a mouse and providing 2D imagery through a monitor. For example, Vaa3D's *Virtual Finger* [51] casts rays through the volume to find the potentially selected objects as the user draws a line with the mouse. Thus users may need to perform multiple interactions and camera rotations to find and select the desired feature, to work around occluders or ambiguous hits in the ray casting process.

3.3 Immersive Environments

There has been a growing interest in using virtual reality or immersive environments for neuron tracing and visualization in general to overcome the limitations of traditional 2D desktop interaction and visualization modalities. Existing tools such as Vaa3D have announced early VR system support, and other new VR-specific tools have been released [1, 56]. In contrast to desktop software, VR and immersive systems allow users to visualize and interact with their data directly in 3D, providing a more intuitive interface and allowing for better understanding of 3D structures [19, 32, 33].

Usher et al. [56] proposed a virtual reality system for manual neuron tracing. In their evaluation, they found that domain experts could perform similar quality traces to standard desktop software in less time, achieving a roughly 2× speedup. Moreover, they found that experts reported the VR tool to be more intuitive and less fatiguing, with the immersive visualization aiding their understanding of the data. However, their tool only supports manual tracing and thus, while faster than working on a desktop, would still require a significant amount of time to trace large data sets.

Immersive systems have also been proposed for visualization of wide-field microscopy data, Boorboor et al. [6] proposed a data processing and feature extraction pipeline, the output of which could be visualized in an immersive display wall visualization system implemented with Unity. Sicat et al. [53] presented DXR, a Unity based toolkit for easily developing immersive visualization applications. Fulmer et al. [20] presented a web-based immersive neuron visualization system using Unity to explore online databases of neuron data in a Hololens.

TOPOLOGY GUIDED NEURON TRACING

In this section, we describe in detail the topological framework developed as the basis for semi-automatic neuron tracing, and the design considerations when integrating this method into a virtual reality neuron tracing system.

4.1 Computing Ridge Graphs

In the images produced through the fluorescence microscopy imaging process described in Sect. 3.1, high-intensity values correspond to the labeled soma, dendrites, and axons, which form the structure of each neuron. When tracing these structures manually, the user aims to produce a path which follows the center-line of these ridge-like structures. Our approach in this work is to generate *every possible ridge-like path* first, turning the neuron reconstruction task into a sub-selection task which can be performed quickly by users. This is in sharp contrast to existing semi-automatic and automatic methods, which attempt to mimic the manual extraction process by computing the single most-likely path for the user.

Our first task is to extract the set of all possible ridge lines from the scalar field. Historically, ridge lines have been defined with techniques relating to the alignment of the principle directions of curvature and the gradient, Eberly et al. [14] provide an excellent overview. However, locally defined ridge lines have major limitations for the task of acting as an acceleration structure for neuron reconstruction. Height ridges do not necessarily form an interconnected network, with segments ending where the local image no longer looks like a ridge. Furthermore, pruning ridge lines by intensity further disconnects the network, exacerbating the problem.

Instead, we use a topological approach that identifies ridge-like structures which are close enough to true ridge lines, but easier to compute. Our use of the MS complex is motivated by the observation that the ridge-like structures formed by the 1-skeleton of the MS complex, composed of the arcs between 2-saddles and maxima, correspond to the center lines of the vast majority of neurons in the data (Fig. 4.1). We then leverage persistent homology (Sect. 2.2.1) to produce a simplified representation of the 1-skeleton of the MS complex.

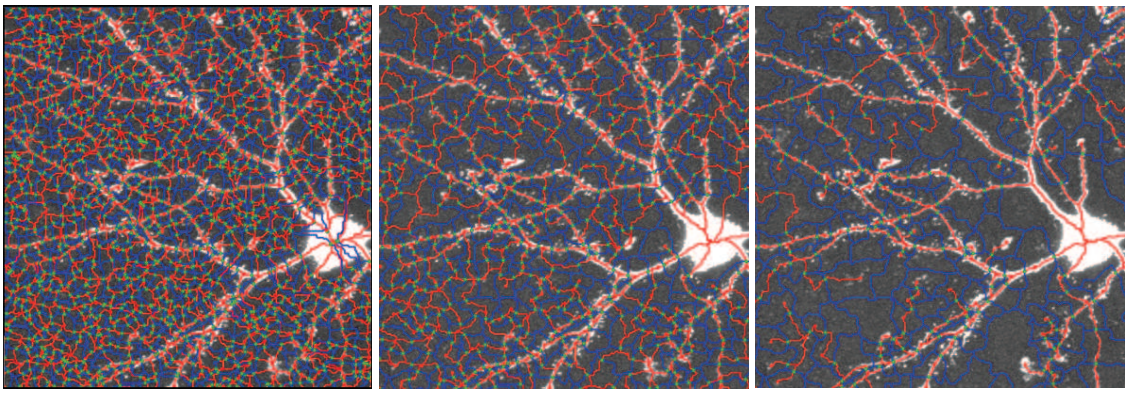
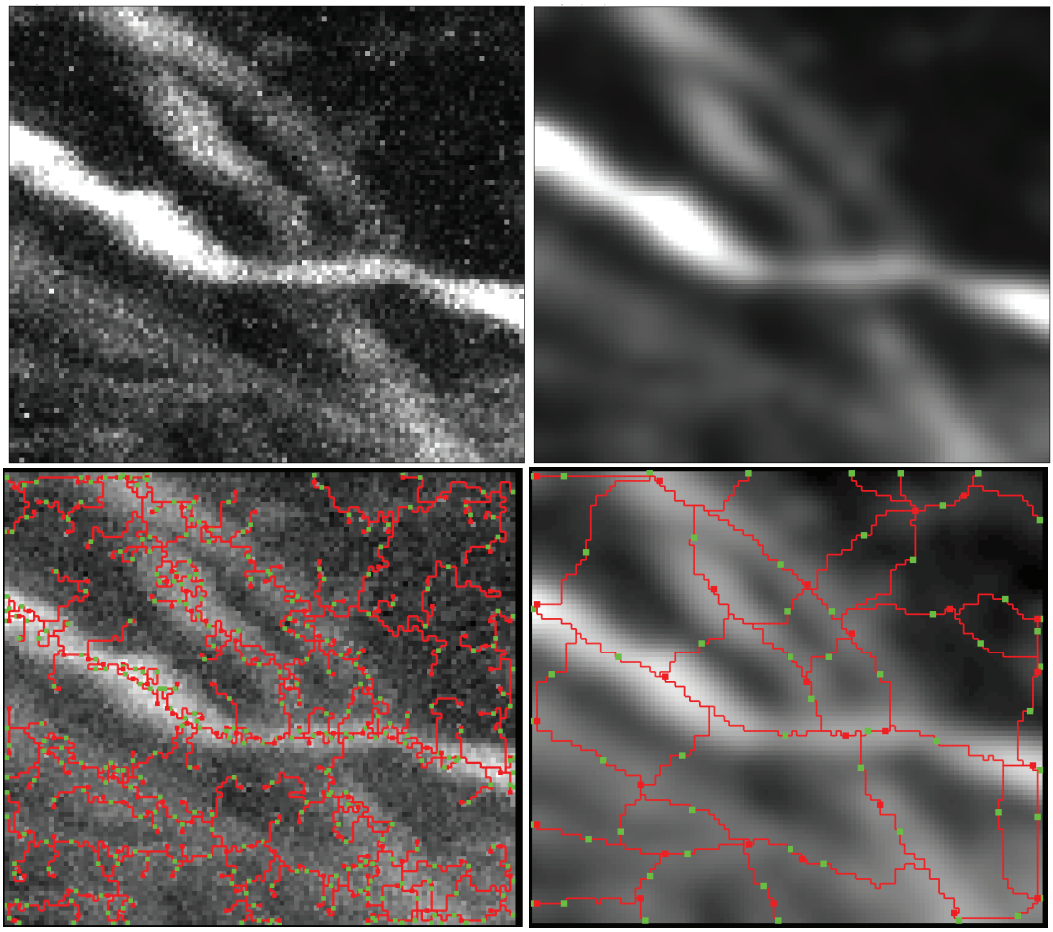


Figure 4.1: From left to right: the full MS complex and its simplification at different persistence thresholds. Red lines represent 2-saddle-maximum arcs and blue lines represent 1-saddle-minimum arcs. The *ridge graph* is composed of the red arcs.

The subset of the 1-skeleton which includes just the maxima, 2-saddles, and the arcs connecting them after simplification provides the desired set of ridge-like structures. We refer to this subset as the *ridge graph*. Fig. 4.1 illustrates the full MS complex and its successive coarsening through persistence simplification. The steps for computing the ridge graph are: image preprocessing, MS complex computation and persistence simplification, and ridge graph post-processing.

Image Preprocessing. A standard approach for working with microscopy data is to apply an image preprocessing step [3], e.g., filtering or blurring, to remove large-amplitude and high-frequency noise. As the MS complex traces gradient trajectories, noisy images will lead to a poor geometric reconstruction of the gradient paths (Fig. 4.2a). We preprocess the input images using a median filter with a radius of 2 and a subsequent Gaussian blur with the same radius, which we found sufficient to remove noise without overblurring. The ridge graph computed on these preprocessed images is sparser and has a higher-quality geometric embedding, while retaining the major neuron structures of interest (Fig. 4.2b).

MS complex Computation and Simplification. We use a standard approach based on discrete Morse theory to construct a discrete gradient field [26, 52], available in the open-source MSCEER [22] library. The library computes a discrete representation of the gradient using a parallel local filter, after which it traces integral paths in the gradient field to construct the 1-skeleton of the MS complex. MSCEER also supports computing the MS complex at a user-specified persistence simplification threshold [30], which we use to simplify out extraneous features created by noise. Higher thresholds will produce



(a) The original image data. (b) After filtering and blurring.

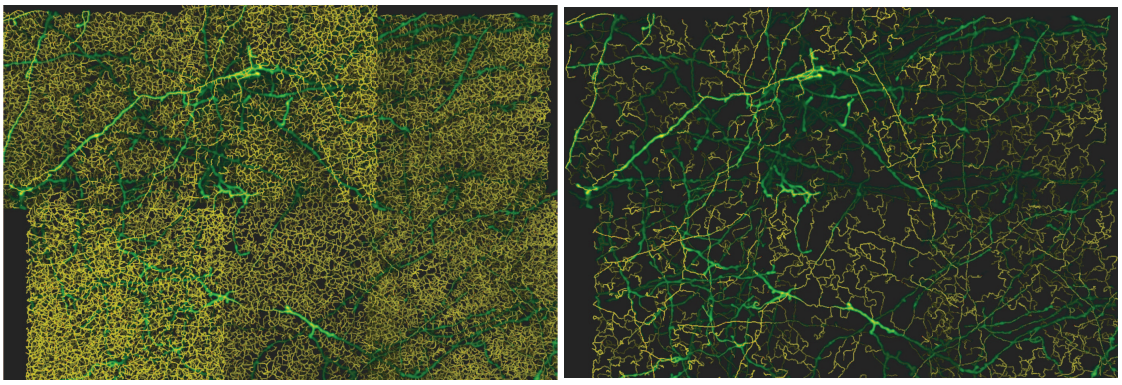
Figure 4.2: The image preprocessing step significantly reduces the effect of noise, allowing for the computation of a more accurate ridge graph.

coarser complexes and sparser ridge graphs; however, selecting too high of a threshold may remove faint but desirable features. We found that selecting a low threshold—as low as 1% of the function range—is sufficient to remove a large portion of the noise while keeping the majority of faint neurons (Fig. 4.3). Finally, the ridge graph is extracted as the 2-saddle-maximum arcs of the MSC.

Ridge Graph Post-Processing. The discrete gradient used in computing the MS complex and ridge graph produces arcs whose segments are aligned to the underlying grid axes. These arcs are smoothed using a simple averaging of neighbor positions to produce more aesthetically pleasing results (Fig. 4.4).

4.2 Virtual Reality Neuron Tracing Framework

We integrate our Morse-Smale Complex guided (MSC-guided) method within an existing virtual reality framework for manual neuron tracing [56] for the HTC Vive. The VR



(a) Persistence threshold near 0. (b) Persistence threshold at 1%.

Figure 4.3: Persistence simplification removes extraneous features due to noise while preserving faint but desirable ones.

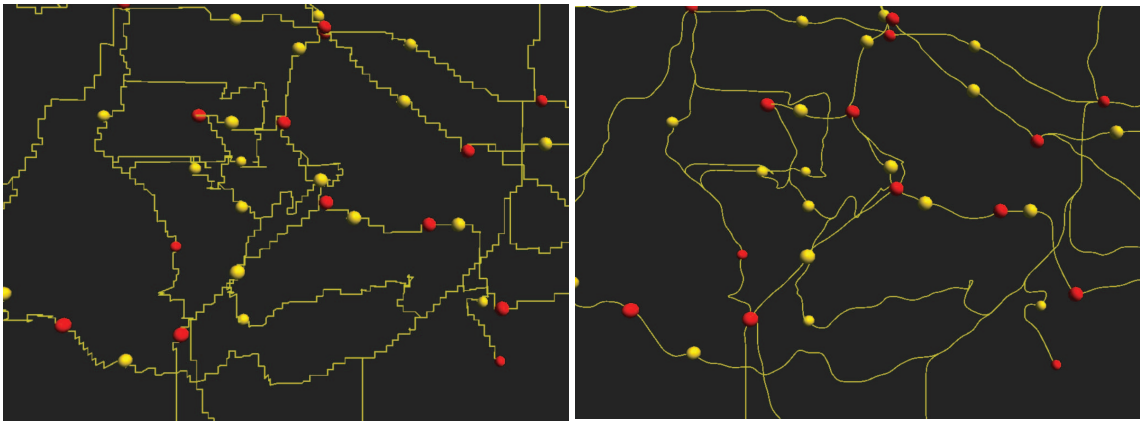


Figure 4.4: The arcs and critical points from the original ridge graph in its discretized form (left), and after applying the smoothing process (right).

framework supports intuitive interaction modes for navigating large volumetric data and manual neuron tracing. Furthermore, the framework supports streaming and rendering large data through the IDX format [45] and integrated caching system, combined with a fast GLSL volume ray caster. We briefly summarize the framework's existing functionality for manual tracing and rendering, then discuss the design and integration of our MSC-guided semi-automatic tracing tool (Sect. 4.3).

4.2.1 Tracing and Navigation

Tracing neurons and navigating the data are the two primary 3D interactions performed when working on a neuron reconstruction, and as such must be quick and intuitive to do. In the VR tool one of the Vive controllers is mapped to tracing, and the other mapped to navigation. Both interactions are initiated by holding the trigger on the respective controller and moving it, to either trace along a structure, or directly grab the volume and translate it. To trace a neuron the user moves the controller through space following

the structure, as a line is painted along it from the tip of the controller. Releasing the trigger then ends the interaction, either stopping the trace or releasing the volume. The user can also navigate by walking in the virtual space, with the navigation interaction used to either stream new data from disk, or reduce the amount of walking needed.

Neurons are branching structures, and thus supporting an intuitive way to trace them is crucial. The user can start a trace off any point on an existing line to begin a new branch off the trace, or follow a branch back to its parent trace and reconnect it. If mistakes are made during the tracing process a quick undo operation can be performed by pressing the trackpad. Corrections can be made when reviewing a trace by selecting portions of the trace with the controller, and deleting them by pressing the trackpad. The user can then re-trace the removed section to correct it.

To assist the user in navigating and tracing the data, a small minimap is displayed to the side of the data set. The minimap shows the bounds of the currently loaded region of data within the entire data set, along with the user's current traces to provide a summary of previously visited regions of the data.

Finally, the tool uses haptic feedback to improve user's perception of selecting branch points or start and end points on existing traces. When hovering the controller near an existing trace close enough that starting a new trace will be connected to it the controller gives a "click" pulse, to give the sensation of having physically selected the object.

4.2.2 Rendering

As scientists may need to use the tool for hours on end, providing a comfortable experience is critical to avoid motion sickness or discomfort. To meet the high resolution and frame rate demands of VR, the framework follows best practices from VR game development [57]. All work on the main thread is tightly budgeted to fit within the 11 ms frame time, with a 3 ms budget left for unexpected interference or costs. The renderer only displays a 256^3 subregion of the volume to keep the volume rendering cost within this time budget, and limits the amount of data paged onto the GPU each frame. The volume is stored in a sparse 3D texture, which is rendered by a standard GLSL volume ray caster. To further reduce the number of pixels (and thus rays) which must be shaded each frame the renderer uses the `NV_clip_space_w_scaling` extension to reduce the rendering

resolution at the edges of the eye, approximating a foveated rendering approach.

4.3 Semi-Automatic Tracing in Virtual Reality

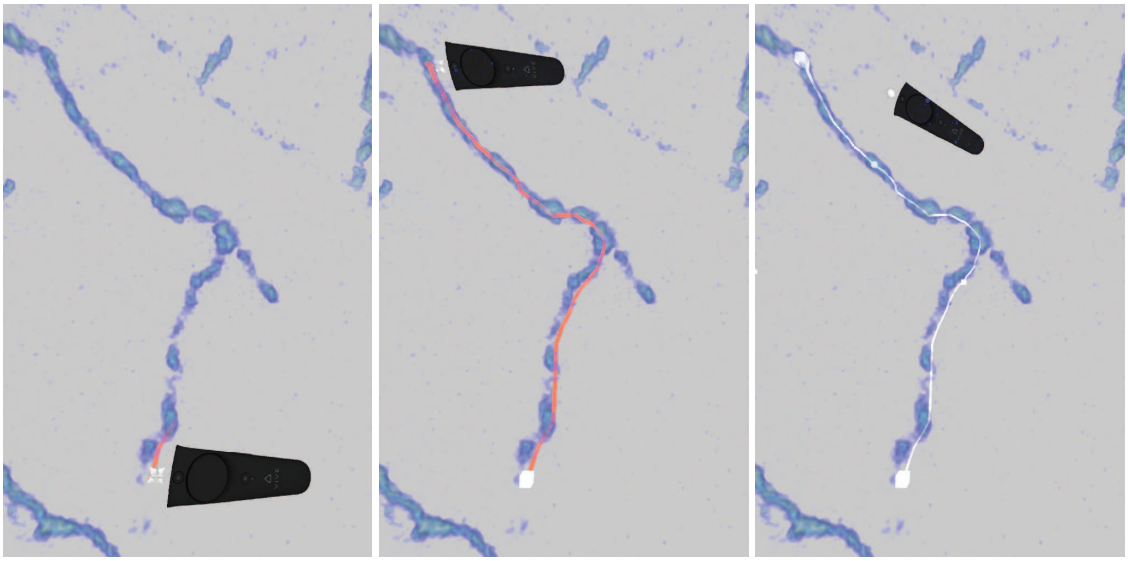
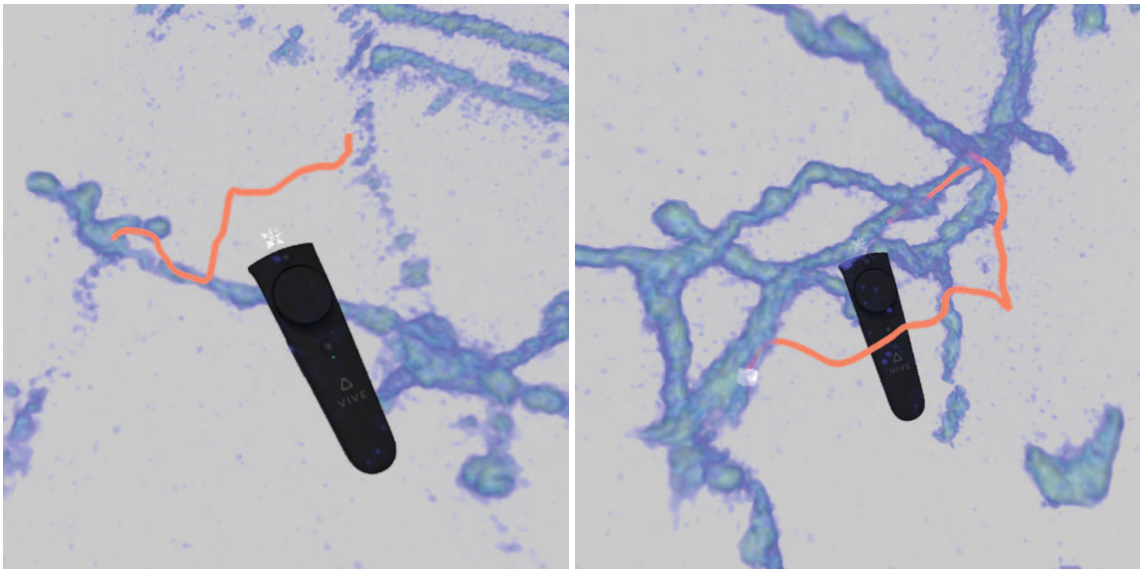


Figure 4.5: Semi-automatic tracing with the MS complex. Left to right: the MS complex live preview shows the hovered segment of a neuron. The user selects this as a start point and moves to the end of the desired segment, viewing the preview (orange). The user then accepts the previewed trace (white).

Our MSC-guided semi-automatic tracing tool allows the user to use the ridge graph to assist in tracing neurons. In designing the tool, our primary focus is on ease of use for neuroscientists, who are unlikely to be familiar with the underlying topological framework. Exposing the MS complex in a way that is intuitive to use for tracing in the VR environment poses some challenges. Displaying the entire set of arcs computed in the ridge graph is distracting and overwhelming (Fig. 4.3a), and may lead to following arcs in the ridge graph which do not correspond to neurons. Instead, we put the data and the neuroscientist's interpretation of it first, and only display the arcs on-demand as the user hovers the tracing controller over regions of the data (Fig. 4.5). To show the on-demand display of the arcs we implemented a unique querying system for the MS complex, discussed in Sect. 4.3.2.

A second issue can arise if the scalar field topology leads the MS complex to follow some other path than the neuron being traced (Fig. 4.6a) or find a shorter path through the ridge graph than is actually desired (Fig. 4.6b). To allow users to quickly work around these cases in our tool, we support switching between the MSC-guided tool and manual tracing. As both tools operate on the same neuron data structure they can be used inter-

changeably as desired by the user. This removes the need to accept a bad trace and return to correct it later, alleviating a common issue with existing semi-automatic and automatic methods. Instead, manual intervention can be done immediately on the fly by switching to the manual tracing tool when necessary.



(a) Features due to background noise. (b) Selecting an incorrect shorter path.

Figure 4.6: In some cases the scalar field topology can lead to incorrect selections made by the MSC-guided tool, due to extraneous features created by background noise, or the tool finding a shorter path through the ridge graph than desired. The live preview is used to view and catch these cases before selecting the trace.

4.3.1 Tracing

When using the MSC-guided tracing tool the arc in the ridge graph which is closest to the controller is highlighted, giving a small live preview of the arcs in the underlying complex. To begin an MSC-guided trace the user places the controller next to the arc they want to start at and presses the trigger to select it. As they move the controller along the neuron being traced, additional arcs are selected from the ridge graph and added to the preview. To end the trace and add the displayed preview to the neuron tree they press the trigger again at the desired end point (Fig. 4.5). By viewing the live preview, users can check that the selected path accurately follows the desired neuron before selecting the trace.

In contrast to tracing manually, where the trigger must be held for the duration of the trace and each neuron segment individually and carefully traced, the two click interaction of the MSC-guided tool provides a significant reduction in fatigue. When using the MSC-

guided tool, the user simply picks a start point and navigates to the desired endpoint, while checking that the preview follows the desired structure. If the preview begins to drift due to the potential issues discussed above, the user can add the set of arcs before the problematic section and continue a new MSC-guided trace or short manual trace off the endpoint.

Capturing the branching structure of each neuron is critical to extracting the connectivity of the neuron, which is used in subsequent analysis tasks. To trace the branches of a neuron, the user can choose between starting a manual trace or an MSC-guided trace to start from some point on the existing trace. When starting an MSC-guided branch, it is possible that no arc exists in the graph to connect the new branch back to the parent tree. In this case, we create a short arc to join the branch with the nearest point on the tree. The branch can then be traced as before using the MSC-guided tool.

4.3.2 A Fast and Efficient Querying Framework

Prior work on semi-automatic neuron tracing is frequently limited by the time it takes to compute the trace when following the user's guidance. The computation can frequently take tens of seconds to minutes, requiring the user to wait before selecting the next section to trace. This wait time impacts productivity on a desktop and is exacerbated in VR, where waiting for computation to complete while blocking the UI or other interactions is unacceptable. To ensure our MSC-guided tracing tool is comfortable to use we impose a hard requirement that it work in real-time with no noticeable frame rate drops. The benefits of imposing this requirement are two-fold: avoiding dropped frames is critical to providing a comfortable experience in VR, and the immediate feedback provided makes the tool faster and easier to use.

We use a k-d tree to quickly select start and end points from the ridge graph. Each arc in the MS complex is discretized into smaller line segments (Fig. 4.4 left), and we store the start and end points that compose these segments in a k-d tree. To select a point in the ridge graph, we query the k-d tree to find the point which is closest to the tracing controller. By querying just the nearest point without any restriction on query radius, users can easily pick start and end points in the ridge graph without having to click exactly on the desired point in space, avoiding the need for precise interactions and thereby reducing fatigue.

The MS complex ridge graph is a connected graph. The vertices are represented by the MS complex 2-saddle and maxima nodes, and edges are represented by the 2-saddle-maxima arcs of the MSC. To compute the preview trace or selection between a designated start and end point in this graph we use Dijkstra's shortest path algorithm [13] between the vertices of the arcs that contain the designated start and end points. To allow for finer grained previews and selections, we allow for starting and ending at points within arcs, to select subsets of the arcs themselves.

It is worth noting that this selection framework could be implemented in a desktop environment, by either querying arc points projected to the screen in 2D, or performing a ray cast into the scene to find the most likely 3D selection. While the topological framework and querying method would remain the same, the user's ability to interact with the data would be impacted. Due to the limitations of performing selections on 3D structures in a 2D desktop environment, we chose to evaluate our method only in VR.

EVALUATION

To evaluate the effectiveness of our topology guided semi-automatic neuron tracing method we study both the effectiveness of the underlying topological framework, and the design of our semi-automatic tracing tool in VR. First, to demonstrate that the Morse-Smale Complex provides an effective framework for neuron tracing we perform an offline comparison against semi-automatic methods available in current desktop software (Sect. 5.1). We then evaluate our MSC-guided tracing tool in virtual reality through a pilot study with trained neuroanatomists and undergraduates, collecting extensive quantitative (Sect. 5.2) and qualitative feedback (Sect. 5.3).

Data and Reference Traces. We evaluate our approach on the *Neocortical Layer 1 Axons* data set [12] made publicly available for the DIADEM challenge [10]. The data set is a $1464 \times 1033 \times 76$ volume made from six aligned subvolumes containing 34 axons imaged from a mouse brain. The resolution of the data is $\approx 0.08\mu m$/pixel in X and Y and $\approx 1\mu m$/pixel along Z. The data set includes a reference trace for each neuron, which we use as one point of comparison in our evaluation. The reference traces were used for comparison in the DIADEM challenge and produced manually using NeuroLucida. Throughout the text we will refer to these traces as the "DIADEM traces." To provide a second point of comparison, we also compare against traces created manually by an expert

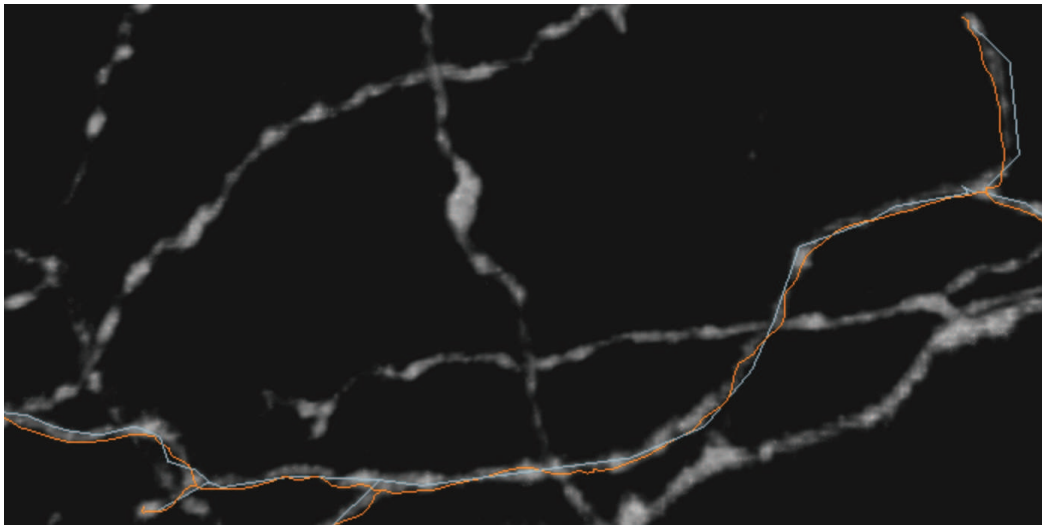


Figure 5.1: The DIADEM reference (blue) was made using standard desktop software, and consists of coarse line segments. The VR reference (orange) consists of finer segments, and follows the neuron more closely.

in VR during a previous study [56], referred to as the "VR reference traces." As neuron traces are produced by hand by experts, there is an inherent subjectivity in each trace, and no real "ground truth" available to compare against. During a review of the provided DIADEM traces we observed that they would often drift from the neuron, following a more linear path than the underlying data, while the VR reference traces followed the structure more closely (Fig. 5.1).

Comparison Metrics. We use two metrics to evaluate the quality of traces produced using our MSC-guided tracing tool. The DIADEM metric [21] accounts for the length and connectivity of a trace, scoring how well a trace captures the branch points and branching structure of a neuron on a scale of 0 (dissimilar) to 1 (identical). The score is penalized for missing branches, excess branches, incorrectly placed branches, and differences in branch length. DIADEM scores correlate well with expert judgement and provide a reasonable proxy for accuracy; however, it does not account for geometric differences.

To score the geometric quality of a trace we use the Fréchet distance. The Fréchet distance is a similarity measure for comparing curves which takes into account the spatial distance between the curves. The distance measured is the minimum maximal distance between points on the two curves. As the Fréchet distance operates on curves, not trees, we compare two neurons by matching the arcs making up the trees based on their start and end points. Arcs are matched based on a search radius to find those starting and stopping at the same location, in which case they are considered to be tracing the same feature.

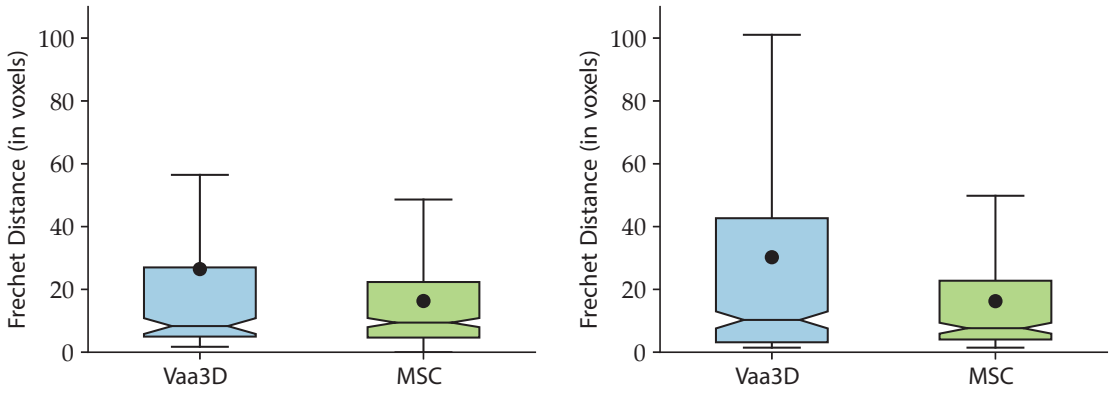
5.1 Offline Comparison

In the offline evaluation we focus on assessing the quality of the neurons computed by each method and their computation time, with the aim of evaluating how well the ridge graph can serve as a framework for tracing neurons. We perform our comparison against Vaa3D's semi-automatic neuron tracing method [47, 50]. Vaa3D is a widely used open-source software suite for neuron reconstruction, and is the designated platform for testing algorithms in the Big Neuron Project [46]. Vaa3D's semi-automatic tracing works similar to our MSC-guided tool: given a start and end point it will attempt to trace the neuron between these two points.

We generate the guide points for each method to trace between by extracting the start,

Reference Trace	Vaa3D	MSC-guided	Improvement
DIADEM	0.45 ± 0.37	0.74 ± 0.26	$1.64\times$
VR	0.53 ± 0.38	0.81 ± 0.22	$1.53\times$

Table 5.1: DIADEM scores for traces extracted with Vaa3D and our MSC-guided method on the reference traces. We find that the MSC-guided method computes better and more consistent traces on average.



(a) Compared to the DIADEM reference. (b) Compared to the VR reference.

Figure 5.2: The Fréchet distances for traces extracted with Vaa3d's semi-automatic method and our MSC-guided approach on the reference traces. The MS complex provides better and more consistent quality traces compared to existing methods

branch, and end points from each neuron in the DIADEM and VR reference traces. Our MSC-guided method works just between a single start and end point, thus we trace the entire neuron by performing a depth first traversal to produce the individual start and end points of each segment. This traversal gives a good approximation of how a user would interact with a semi-automatic method, by clicking along the structure to mark key points along the neuron and letting the algorithm extract the structure. Along with supporting single start and end point extraction, Vaa3D can trace from a start point to connect to a set of points, we use this mode in our evaluation.

When comparing trace quality, we find that those computed by the MSC-guided method follow the neuron more accurately than those computed by Vaa3D's semi-automatic method. We compute the DIADEM scores of each method compared to the respective reference trace used to create the guide points (Table 5.1). Our MSC-guided method captures the branching structure of the neurons more accurately, corresponding to a $1.58\times$ higher score overall. This can be partly attributed to the MSC-guided method placing greater weight on the user's guidance, extracting the shortest path between the given end points as they are clicked in order. In Vaa3D's semi-automatic method, the landmarks serve more as hints and the algorithm is not guaranteed to directly connect them in order.

When comparing the geometric accuracy of the methods, we find the MSC-guided

method follows the desired neuron structure more closely than Vaa3D's semi-automatic method (Fig. 5.2). Across both reference traces Vaa3D's semi-automatic method achieves an average Fréchet distance of 28.5 voxels, while the MSC-guided method averages 16.3 voxels, providing a 1.75× improvement. We note that on the DIADEM reference traces Vaa3D's semi-automatic method failed to trace 9 of the 34 neurons.

Although the MSC-guided method achieves significant improvement in accuracy and consistency, the absolute accuracy could be improved further. The average diameter of a neuron in the data set is roughly 5–8 voxels, and thus a deviation beyond this corresponds to a significant error which would require correction. When using the MS complex for guidance in VR this issue is alleviated, as the user can perform corrections on the fly while tracing.

Our MSC-guided approach is also able to provide much greater interactivity by significantly reducing the trace computation time. We found that on average Vaa3D took 28.64s per-neuron, while our MSC-guided method took just 0.029s, achieving a speedup of 986×. The MSC-guided method computes the ridge graph in a preprocess to provide the set of candidate arcs which the user selects from at runtime, reducing the time they must wait for the algorithm when using the tool. Moreover, the precomputation to build the ridge graph is fast and scalable [22, 26]. On a laptop with an i7–7700HQ CPU the image filtering and blurring takes 26s in ImageJ, after which the MS complex is computed in 134s using MSCEER [22].

5.2 Expert Evaluation

In our expert evaluation we focus on the usability of the tool by experts in practice through a pilot study (IRB_00099920). We conduct our study with five users with varying levels of experience from A. A.'s laboratory. Subjects 2 and 5 are senior neuroanatomists, subjects 1 and 4 are undergraduate students with 2–3 years of experience tracing neurons using NeuroLucida, and subject 3 is a novice with little prior experience. This range of experience levels provides a representative sample of a typical connectomics lab, where senior researchers train inexperienced undergraduates who are hired to do the bulk of the neuron tracing work.

To compare our MSC-guided tracing method against a fully manual trace we have each

User	Manual Score	MSC-guided Score	Improvement
Compared to DIADEM Reference			
1	0.53 ± 0.38	0.66 ± 0.35	$1.25\times$
2	0.57 ± 0.36	0.60 ± 0.37	$1.06\times$
3	0.39 ± 0.35	0.53 ± 0.37	$1.35\times$
4	0.27 ± 0.32	0.46 ± 0.36	$1.69\times$
5	0.35 ± 0.35	0.38 ± 0.36	$1.10\times$
Compared to VR Reference			
1	0.66 ± 0.32	0.57 ± 0.35	$0.86\times$
2	0.66 ± 0.34	0.58 ± 0.40	$0.88\times$
3	0.55 ± 0.55	0.52 ± 0.52	$0.94\times$
4	0.29 ± 0.29	0.53 ± 0.53	$1.81\times$
5	0.49 ± 0.49	0.46 ± 0.46	$0.94\times$

Table 5.2: Average DIADEM scores and standard deviation compared to the DIADEM and VR reference traces. We find users perform similar or slightly better quality traces when using the MSC-guided tool.

subject trace the set of neurons twice over two separate sessions, spaced at least five days apart. The first two neurons are used for a short training session to introduce the MSC tool and VR environment, and the remaining 32 for evaluation. For each neuron the start point is marked in space and the user is instructed to trace the neuron to its perceived end points. In each session half the neurons are traced manually and half using any combination of the MSC-guided tool and manual tracing as desired by the user. In the second session the set of neurons which are traced manually or with the MSC-guided tool is flipped. On average, each session took an hour to an hour and a half.

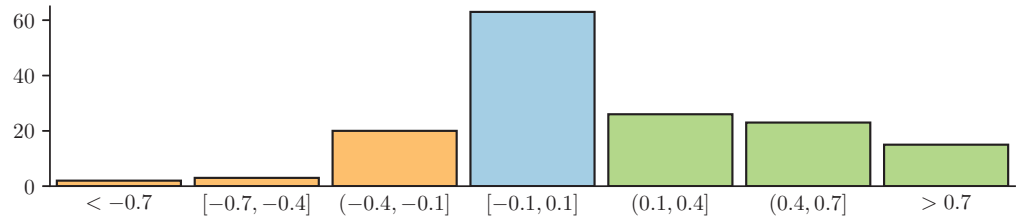
We find that the quality of the traces produced when using fully manual tracing and our MSC-guided semi-automatic tracing are similar (Table 5.2). When computing DIADEM scores against the DIADEM reference traces we see a moderate improvement for some users; however, the scores for each tool are within standard deviation of each other. The DIADEM scores computed against the VR reference traces are similar, with a slight decrease in score observed for most users. When comparing the difference in score achieved using the MSC-guided tool against manual tracing for each neuron, we find that the majority of traces are within acceptable error of each other, with traces slightly more likely to be better when using the MSC-guided tool (Fig. 5.3). We find similar results when comparing the Fréchet distance, and see no significant difference in accuracy when using the MSC-guided tool (Fig. 5.4).

It is interesting to note that the average DIADEM score achieved by the MS complex in

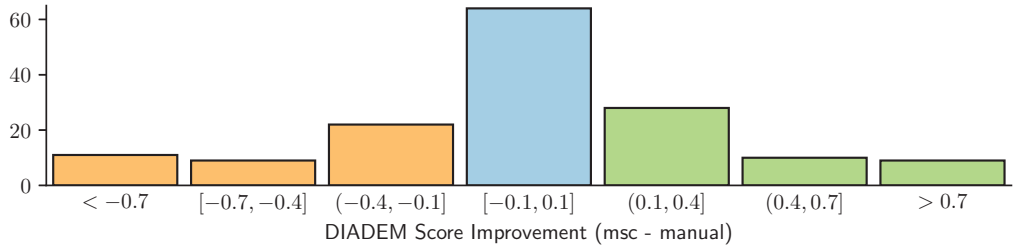
the offline comparison is higher than that achieved by any user in the evaluation. When performing the offline comparison the algorithm is given the entire set of start, branch and end points, and thus cannot miss a branch; however, the experts are only given the start point and could miss branches, resulting in a lower DIADEM score. Compared to the offline MS complex traces the experts achieve much lower Fréchet distances, indicating that the portions of the neuron which are traced are done so with greater accuracy. When using the MSC-guided tool experts can catch and correct errors in the extracted arcs to produce a better trace, while the offline comparison cannot.

When comparing the time spent tracing with the manual tool and the MSC-guided tool the results initially seem disappointing, with little speedup achieved when using the MSC-guided tool on average (Table 5.3). However, if we compute the speedup achieved using the MSC-guided tool separately in each session we find that three of the five subjects achieve a significant speedup in the second session, but little to none in the first (Table 5.4). This result indicates that the MSC-guided tool has a higher learning curve than we initially anticipated. Users may feel less confident using the MSC-guided tool during the first session, and spend more time second guessing it or tracing manually. We collected additional qualitative feedback during the survey and discussions which support this hypothesis (see Sect. 5.3).

It is also worth noting that the tracing time is inherently dependent on the expert's

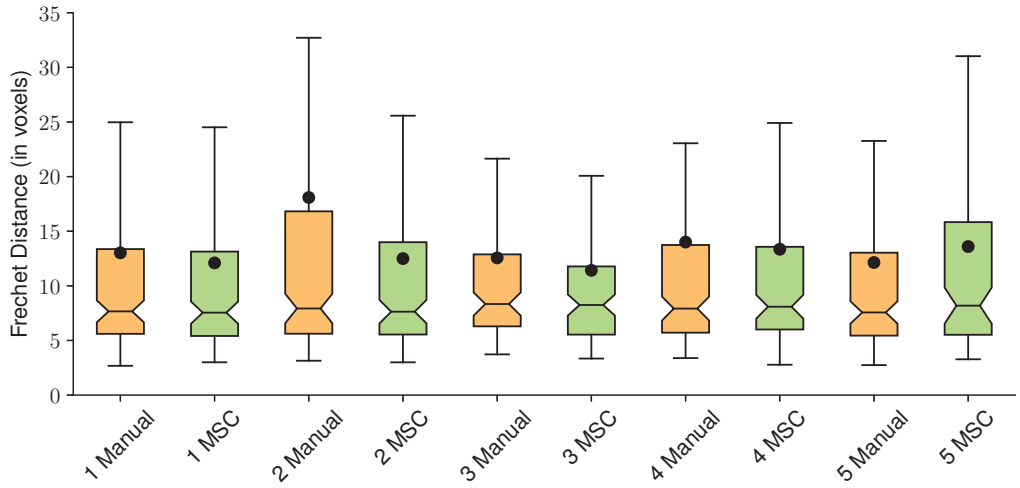


(a) Compared to the DIADEM reference.

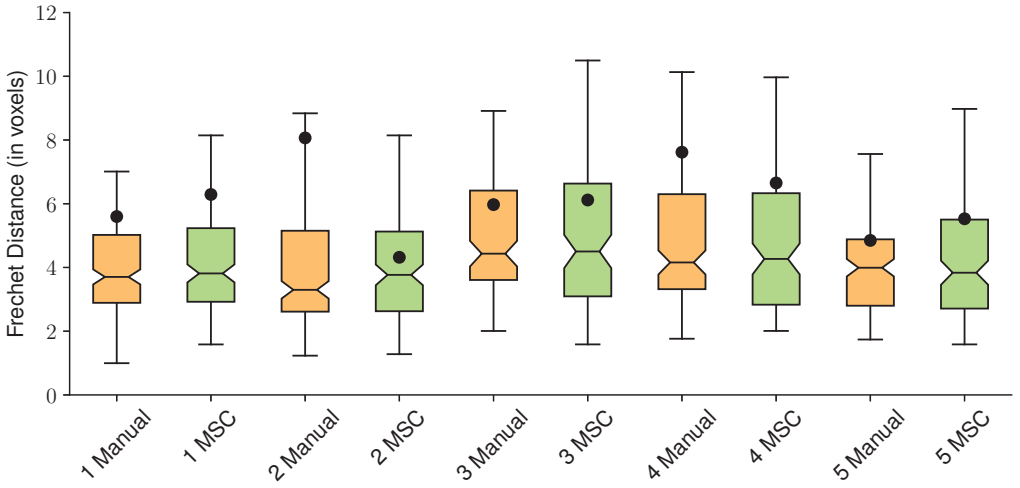


(b) Compared to the VR reference.

Figure 5.3: The difference in score when using the MSC-guided method vs. manual tracing, across all users. The majority of traces using the MSC-guided tool are similar to the manual traces, or slightly better.



(a) Compared to the DIADEM reference.



(b) Compared to the VR reference.

Figure 5.4: Fréchet distances for user manual and MSC-guided user traces. We find that users perform similar quality traces using both tools.

judgement. For example, if in one session the user misses a large branching structure and traces it in the next, the second session will take longer. When testing to determine if speedup achieved using the MSC-guided tool correlates with lower DIADEM scores, we do not find a correlation (Fig. 5.5, Pearson correlation coefficient 0.28). This indicates that the speedup achieved using the MSC-guided tool is not due to performing incorrect, shorter traces than when tracing manually.

5.3 Discussion

In this section we detail our users' qualitative feedback regarding the design and usability of the MSC-guided tracing tool, discussing both benefits and limitations. During the study we collected feedback from the users through a survey completed after each

User	Manual (s)	MSC-guided (s)	Speedup
1	154.44	154.06	1.00×
2	271.83	222.35	1.22×
3	94.78	131.34	0.72×
4	134.67	108.73	1.24×
5	235.38	167.09	1.41×

Table 5.3: Average tracing times for manual tracing and our MSC-guided tool across all sessions.

session and open-ended discussions. The survey focused on the usability and usefulness of the MSC-guided tool for tracing neurons, with questions rated on a 5-point Likert scale. The open-ended discussion solicited general feedback on the design of the tool and general comments or issues regarding using it in practice.

User Experience Overall feedback from users on the MSC-guided tool was positive. In the survey all subjects reported preferring the MSC-guided tool over manual tracing, finding it less fatiguing, although more difficult to use at first. Subject 4 mentioned feeling more comfortable using the MSC tool, as it required paying less attention to closely tracing each neuron. Users reported that the ability to quickly switch between the manual and MSC-guided tools was valuable to pick the right tool for the task at hand. When resolving

User	1st Session Speedup	2nd Session Speedup
1	1.04×	0.91×
2	1.28×	1.79×
3	1.02×	0.85×
4	0.94×	2.29×
5	0.83×	2.98×

Table 5.4: A comparison of the speedup achieved using the MSC-guided tool compared to manual tracing in the first vs. the second session. In the second session 3 of 5 users achieved significantly higher speedups than the first, indicating a higher learning curve than was originally anticipated.

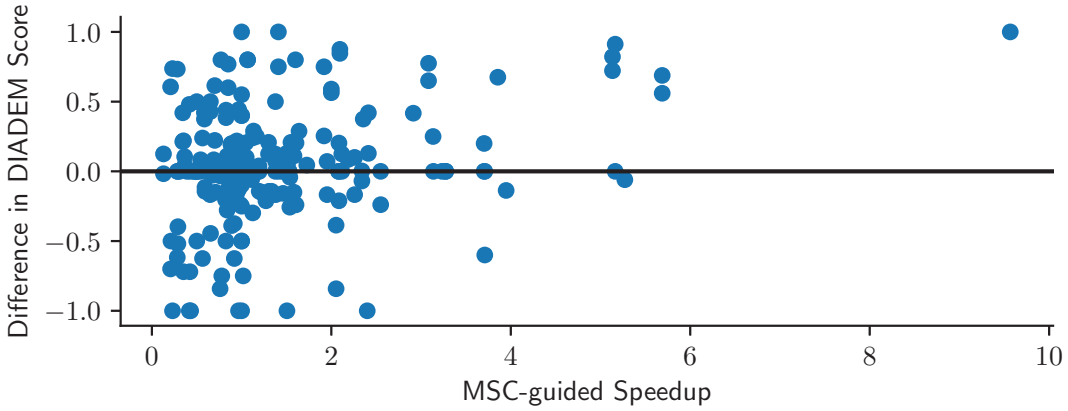


Figure 5.5: We find no correlation between speedup when using the MSC-guided tool and lower DIADEM scores, indicating that the speedup is not the result of performing faster, lower quality traces.

complex crossings or tracing through regions where the MS complex did not follow the desired path they would switch to the manual tool, and switch back to the MSC-guided tool after finishing the section.

Users found the MSC-guided tool's live trace preview useful to view the trace which will be selected before actually selecting it. The preview enables users to catch regions where the MS complex struggles and switch to the manual tool to trace the section, instead of having to go back and validate each segment traced with the MSC-guided tool after the fact. The preview helps users trace the data faster by significantly reducing the amount of proof-editing required of the semi-automatic traces, as the editing task is integrated into the semi-automatic tracing task itself. In contrast, existing semi-automatic methods do not provide a live preview as the computation is often too slow to do so, and users must instead accept and subsequently correct the trace created by the algorithm.

Manual vs. MSC-guided Tracing. When comparing the two tools, the majority of users reported finding the MSC-guided tool more challenging to learn. A frequent comment from users was that it took a few traces to become comfortable with the MSC-guided tool, and to learn the cases where it would accurately trace the neuron and where it would fail. After learning these cases they trusted the MSC-guided tool more, as they could predict its behavior in various situations and either quickly mark start and end points or switch to the manual tool where necessary. One subject commented that it was only in their final traces of the second session where they felt they truly took advantage of the MSC-guided tool to accelerate their tracing. Subject 3 reported preferring manual tracing after the first session, though reversed this preference after the second session.

Users reported the tool to be especially useful when tracing long axons through large portions of the volume. When using the MSC-guided tool they would let it follow the neuron for them, and focus on navigating to the end point of the axon to finish the trace. When tracing manually this task is more difficult, as users must typically swap between tracing and navigating to create an accurate trace. During the manual portion of the second session, many subjects lamented not being able to use the MSC-guided method to trace these sections.

There are a number of regions in the data where a neuron may appear to end or fade due to issues with the tissue labeling or imaging process. All subjects reported that the

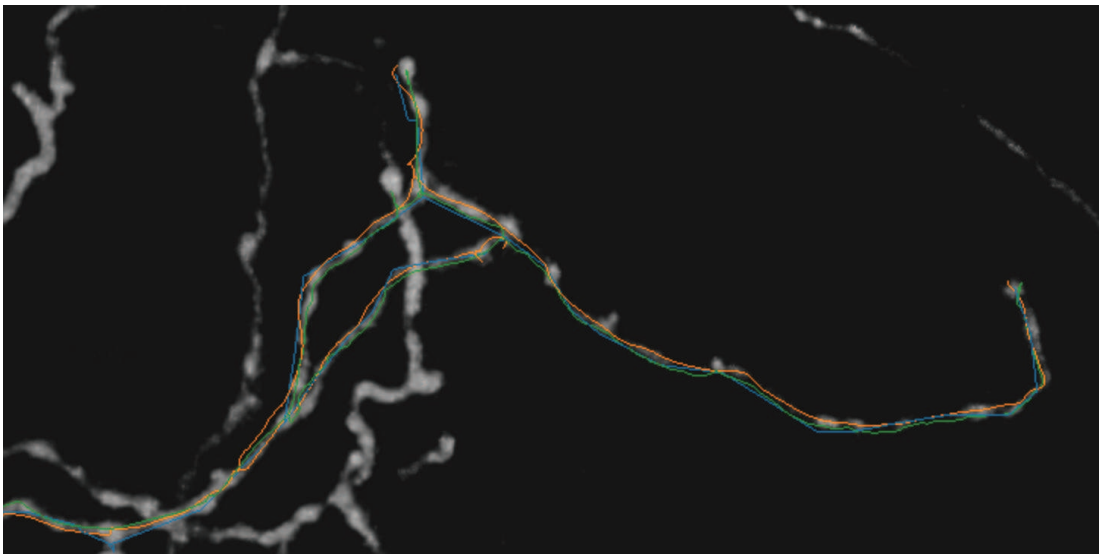


Figure 5.6: The accuracy differences between an MS complex trace (green) and the corresponding DIADEM (blue) and VR (orange) reference traces. Our MSC-guided tool helps the user better follow the neuron center line to produce an accurate trace.

MSC-guided tool was particularly helpful in resolving these portions of the data. One of the senior neuroanatomists, subject 2, reported that using the MSC-guided tool helped him analyze these cases more carefully. On one trace this led to him ultimately determining that a neuron did not end where he initially thought it did. Subjects 4 and 5 made similar comments, noting that the MSC-guided tool helped them make decisions at branch points and potential termination points.

When evaluating the traces individually we found some instances where users failed to correctly connect their traces, either when creating a branch or continuing off an existing trace. Across all subjects, we found 24 manual traces and 15 MSC-guided traces with at least one missed connection. These can occur when the user intends to start a branch but does not quite select the trace to branch from, or stopped tracing to move the volume and does not restart off the original trace, by slightly missing the end point when continuing their trace. The latter case is less likely to occur when using the MSC-guided tool, as users can use it to trace the neuron for them while they navigate the volume, as discussed previously. Providing clearer feedback when branching or creating a disconnected trace would be desirable to further reduce these missed connections.

Finally, during a visual inspection of the traces we found that in many cases the MSC-guided trace followed the neuron more closely than the DIADEM and VR reference traces (Fig. 5.6). The DIADEM trace clearly shows the limitations of tracing manually on the

desktop, where users click and place points to construct straight line segments between them. In VR both the manual and MSC-guided tools create a finer line which can follow the structure more accurately, as the tools behave more similar to a paint brush. However, when tracing manually in VR users can have difficulty tracing exactly along the center line of the neuron. The MSC-guided tool alleviates this issue by automatically following the ridgeline of the neuron for the user, requiring them to only provide a coarse set of inputs.

SUMMARY AND FUTURE WORK

We have presented a novel semi-automatic neuron tracing method based on the topological framework of the Morse-Smale Complex. We implemented our Morse-Smale Complex guided tracing tool within an existing VR environment, and demonstrated that it improves neuron tracing performance over manual tracing in VR and semi-automatic methods on the desktop. When using our MSC-guided tracing tool, experts were able to produce high-quality traces with less fatigue and in less time. By leveraging the fast online computation time of our method, we are able to show a live preview of the trace to the user, removing the need for extensive post-process proof-editing of the trace. Moreover, the neuroanatomists' qualitative feedback indicates that, although more work remains to be done, our MSC-guided tool is a promising approach to accelerate neuron tracing, especially in low resolution regions and when tracing long range connections.

Although the results of our pilot study are promising, we have also found areas for improvement. In our evaluation the domain scientists noted that the MSC tool was more difficult to learn than manual tracing, and in the evaluation we observed little or no speedup when using the tool in the first session. This suggests the need for a more detailed training process when first introducing users to the MSC tool, to help them more quickly learn about its strengths and weaknesses. To make the tool more intuitive for first time users it may be useful to provide a predictive live preview, which shows possible continuations of the path being traced. During the evaluation, subject 2 expressed interest in being shown multiple potential paths which could be chosen to connect the start and end points. A better training process or clearer to use tool would allow new users to become effective with the tool sooner, which is especially important when training new hires.

During the evaluation we also found that in multiple cases users would fail to connect the trace correctly. In total we found this mistake was made in 15% of traces when tracing manually and 9.4% of traces when using the MSC tool. Though the MSC tool does reduce the frequency of this mistake by making it easier to navigate and trace simultaneously, this issue could be further addressed by highlighting these disconnected regions and providing clearer feedback to users when branching or continuing off an existing trace. Subject 1

noted that it would be useful to be able to mark and return to branch points and other areas, reducing the chance that users forget to return to a branch point created early on in the trace.

Another potential avenue for improvement is to evaluate different methods for weighting the ridge graph when computing the shortest path for a trace. For example, weighting the ridge graph by both distance and the value of a critical point, with higher values assigned a lower weight, could resolve cases similar to that in fig:msc-bad-shorter-path. However, finding a good balance between weighting by distance and function value is key to providing an accurate trace. One possibility is to show multiple options, based on different weight balances in the graph, as suggested by subject 2. These weights could be further adjusted on the fly, based on how often one weight ratio is chosen over another to learn the best weights for the data set. Finally, these previews could be weighted by the uncertainty in the ridge graph, if the topological framework is able to provide uncertainty information along with the arcs. Computing uncertain topological structures is currently an open and active field of research.

We are in the process of publicly releasing our MSC-guided tracing tool to make it widely available to neuroscientists. Moreover, a public release of the tool will allow us to explore application of our MSC-guided tracing tool to other tasks and data sets. In addition, this work has been submitted as a report [2].

REFERENCES

[1] syGlass. https://www.syglass.io/.

[2] Technical report uu-sci-002. *SCI Institute University of Utah*.

[3] L. Acciai, P. Soda, and G. Iannello. Automated neuron tracing methods: an updated account. *Neuroinformatics*, 14(4):353–367, 2016.

[4] H. Bhatia, A. G. Gyulassy, V. Lordi, J. E. Pask, V. Pascucci, and P.-T. Bremer. Topoms: Comprehensive topological exploration for molecular and condensed-matter systems. *Journal of computational chemistry*, 39(16):936–952, 2018.

[5] D. D. Bock, W.-C. A. Lee, A. M. Kerlin, M. L. Andermann, G. Hood, A. W. Wetzel, S. Yurgenson, E. R. Soucy, H. S. Kim, and R. C. Reid. Network anatomy and in vivo physiology of visual cortical neurons. *Nature*, 471(7337):177, 2011.

[6] S. Boorboor, S. Jadhav, M. Ananth, D. Talmage, L. Role, and A. Kaufman. Visualization of neuronal structures in wide-field microscopy brain images. *IEEE transactions on visualization and computer graphics*, 25(1):1018–1028, 2019.

[7] P.-T. Bremer, B. Hamann, H. Edelsbrunner, and V. Pascucci. A topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 10(4):385–396, 2004.

[8] P.-T. Bremer, G. Weber, V. Pascucci, M. Day, and J. Bell. Analyzing and tracking burning structures in lean premixed hydrogen flames. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):248–260, 2010.

[9] K. L. Briggman, M. Helmstaedter, and W. Denk. Wiring specificity in the direction-selectivity circuit of the retina. *Nature*, 471(7337):183, 2011.

[10] K. M. Brown, G. Barrionuevo, A. J. Canty, V. De Paola, J. A. Hirsch, G. S. Jefferis, J. Lu, M. Snippe, I. Sugihara, and G. A. Ascoli. The diadem data sets: representative light microscopy images of neuronal morphology to advance automation of digital reconstructions. *Neuroinformatics*, 9(2-3):143–157, 2011.

[11] K. Chung, J. Wallace, S.-Y. Kim, S. Kalyanasundaram, A. S. Andalman, T. J. Davidson, J. J. Mirzabekov, K. A. Zalocusky, J. Mattis, A. K. Denisin, et al. Structural and molecular interrogation of intact biological systems. *Nature*, 497(7449):332, 2013.

[12] V. De Paola, A. Holtmaat, G. Knott, S. Song, L. Wilbrecht, P. Caroni, and K. Svoboda. Cell type-specific structural plasticity of axonal branches and boutons in the adult neocortex. *Neuron*, 49(6):861–875, 2006.

[13] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1959.

[14] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach. Ridges for image analysis. *Journal of Mathematical Imaging and Vision*, 4(4):353–373, 1994.

[15] Edelsbrunner, Harer, and Zomorodian. Hierarchical morse—smale complexes for piecewise linear 2-manifolds. *Discrete & Computational Geometry*, 30(1):87–107, May 2003. doi: 10.1007/s00454-003-2926-5

[16] H. Edelsbrunner and J. Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[17] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Morse-smale complexes for piecewise linear 3-manifolds. *Proceedings of the Annual Symposium on Computational Geometry*, 2003, 04 2003. doi: 10.1145/777792.777846

[18] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. pp. 454–463, 2000.

[19] A. Forsberg, M. Katzourin, K. Wharton, M. Slater, et al. A comparative study of desktop, fishtank, and cave systems for the exploration of volume rendered confocal data sets. *IEEE Transactions on Visualization and Computer Graphics*, 14(3):551–563, 2008.

[20] W. Fulmer, T. Mahmood, Z. Li, S. Zhang, J. Huang, and A. Lu. Imweb: cross-platform immersive web browsing for online 3d neuron database exploration. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pp. 367–378. ACM, 2019.

[21] T. A. Gillette, K. M. Brown, and G. A. Ascoli. The diadem metric: comparing multiple reconstructions of the same neuron. *Neuroinformatics*, 9(2-3):233, 2011.

[22] A. Gyulassy. MSCEER: Morse-Smale Complex Extraction, Exploration, Reasoning. `https://github.com/sci-visus/MSCEER`, 2018.

[23] A. Gyulassy, P.-T. Bremer, B. Hamann, and V. Pascucci. A practical approach to morse-smale complex computation: Scalability and generality. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1619–1626, 2008.

[24] A. Gyulassy, P.-T. Bremer, and V. Pascucci. Computing morse-smale complexes with accurate geometry. *IEEE transactions on visualization and computer graphics*, 18(12):2014–2022, 2012.

[25] A. Gyulassy, P.-T. Bremer, and V. Pascucci. Computing morse-smale complexes with accurate geometry. *IEEE transactions on visualization and computer graphics*, 18(12):2014–2022, 2012.

[26] A. Gyulassy, P.-T. Bremer, and V. Pascucci. Shared-memory parallel computation of morse-smale complexes with improved accuracy. *IEEE transactions on visualization and computer graphics*, 25(1):1183–1192, 2019.

[27] A. Gyulassy, M. Duchaineau, V. Natarajan, V. Pascucci, E. Bringa, A. Higginbotham, and B. Hamann. Topologically clean distance fields. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1432–1439, 2007.

[28] A. Gyulassy, A. Knoll, K. C. Lau, B. Wang, P.-T. Bremer, M. E. Papka, L. A. Curtiss, and V. Pascucci. Interstitial and interlayer ion diffusion geometry extraction in graphitic nanosphere battery materials. *IEEE transactions on visualization and computer graphics*, 22(1):916–925, 2016.

[29] A. Gyulassy, N. Kotava, M. Kim, C. D. Hansen, H. Hagen, and V. Pascucci. Direct feature visualization using morse-smale complexes. *IEEE transactions on visualization and computer graphics*, 18(9):1549–1562, 2012.

[30] A. Gyulassy, V. Natarajan, V. Pascucci, P.-T. Bremer, and B. Hamann. A topological approach to simplification of three-dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics*, 2006.

[31] A. Gyulassy, V. Natarajan, V. Pascucci, and B. Hamann. Efficient computation of morse-smale complexes for three-dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1440–1447, 2007.

[32] B. Laha, D. A. Bowman, and J. J. Socha. Effects of vr system fidelity on analyzing isosurface visualization of volume datasets. *IEEE Transactions on Visualization & Computer Graphics*, (4):513–522, 2014.

[33] B. Laha, K. Sensharma, J. D. Schiffbauer, and D. A. Bowman. Effects of immersion on visual analysis of volume data. *IEEE Transactions On Visualization & Computer Graphics*, (4):597–606, 2012.

[34] D. Laney, A. Mascarenhas, P. Miller, V. Pascucci, et al. Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1053–1060, 2006.

[35] Y. Liu. The diadem and beyond. *Neuroinformatics*, 9(2):99–102, Sep 2011. doi: 10.1007/s12021-011-9102-5

[36] L. Luo, E. M. Callaway, and K. Svoboda. Genetic dissection of neural circuits. *Neuron*, 57(5):634–660, 2008.

[37] MBF Bioscience. NeuroLucida 360.

[38] MBF Bioscience. NeuroLucida version 2019.

[39] E. Meijering. Neuron tracing in perspective. *Cytometry Part A*, 77(7):693–704, 2010.

[40] E. Meijering, M. Jacob, J.-C. Sarria, P. Steiner, H. Hirling, and M. Unser. Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry Part A: the journal of the International Society for Analytical Cytology*, 58(2):167–176, 2004.

[41] J. Milnor. *Morse Theory*. Princeton Univ. Press, 1963.

[42] E. Murray, J. H. Cho, D. Goodwin, T. Ku, J. Swaney, S.-Y. Kim, H. Choi, Y.-G. Park, J.-Y. Park, A. Hubbert, et al. Simple, scalable proteomic imaging for high-dimensional profiling of intact systems. *Cell*, 163(6):1500–1514, 2015.

[43] D. Myatt, T. Hadlington, G. Ascoli, and S. Nasuto. Neuromantic–from semi-manual to semi-automatic reconstruction of neuron morphology. *Frontiers in neuroinformatics*, 6:4, 2012.

[44] S. W. Oh, J. A. Harris, L. Ng, B. Winslow, N. Cain, S. Mihalas, Q. Wang, C. Lau, L. Kuan, A. M. Henry, et al. A mesoscale connectome of the mouse brain. *Nature*, 508(7495):207, 2014.

[45] V. Pascucci and R. J. Frank. Global static indexing for real-time exploration of very large regular grids. In *SC'01: Proceedings of the 2001 ACM/IEEE Conference on Supercomputing*, pp. 45–45. IEEE, 2001.

[46] H. Peng, M. Hawrylycz, J. Roskams, S. Hill, N. Spruston, E. Meijering, and G. A. Ascoli. Bigneuron: large-scale 3d neuron reconstruction from optical microscopy images. *Neuron*, 87(2):252–256, 2015.

[47] H. Peng, F. Long, and G. Myers. Automatic 3d neuron tracing using all-path pruning. *Bioinformatics*, 27(13):i239–i247, 2011.

[48] H. Peng, F. Long, T. Zhao, and E. Myers. Proof-editing is the bottleneck of 3d neuron reconstruction: the problem and solutions. *Neuroinformatics*, 9(2-3):103–105, 2011.

[49] H. Peng, E. Meijering, and G. A. Ascoli. From DIADEM to BigNeuron. *Neuroinformatics*, 13(3):259–260, Jul 2015. doi: 10.1007/s12021-015-9270-9

[50] H. Peng, Z. Ruan, F. Long, J. H. Simpson, and E. W. Myers. V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets. *Nature biotechnology*, 28(4):348, 2010.

[51] H. Peng, J. Tang, H. Xiao, A. Bria, J. Zhou, V. Butler, Z. Zhou, P. T. Gonzalez-Bellido, S. W. Oh, J. Chen, et al. Virtual finger boosts three-dimensional imaging and microsurgery as well as terabyte volume image visualization and analysis. *Nature communications*, 5:4342, 2014.

[52] V. Robins, P. J. Wood, and A. P. Sheppard. Theory and algorithms for constructing discrete morse complexes from grayscale digital images. *IEEE Transactions on pattern analysis and machine intelligence*, 33(8):1646–1658, 2011.

[53] R. Sicat, J. Li, J. Choi, M. Cordeil, W.-K. Jeong, B. Bach, and H. Pfister. Dxr: A toolkit for building immersive data visualizations. *IEEE transactions on visualization and computer graphics*, 25(1):715–725, 2019.

[54] L. Silvestri, A. Bria, L. Sacconi, G. Iannello, and F. Pavone. Confocal light sheet microscopy: micron-scale neuroanatomy of the entire mouse brain. *Optics express*, 20(18):20582–20598, 2012.

[55] T. Sousbie. The persistent cosmic web and its filamentary structure–i. theory and implementation. *Monthly Notices of the Royal Astronomical Society*, 414(1):350–383, 2011.

[56] W. Usher, P. Klacansky, F. Federer, P.-T. Bremer, A. Knoll, J. Yarch, A. Angelucci, and V. Pascucci. A virtual reality visualization tool for neuron tracing. *IEEE transactions on visualization and computer graphics*, 24(1):994–1003, 2018.

[57] A. Vlachos. Advanced VR Rendering. GDC, 2015.

[58] B. Yang, J. B. Treweek, R. P. Kulkarni, B. E. Deverman, C.-K. Chen, E. Lubeck, S. Shah, L. Cai, and V. Gradinaru. Single-cell phenotyping within transparent intact tissue through whole-body clearing. *Cell*, 158(4):945–958, 2014.