NeuroLines: A Subway Map Metaphor for Visualizing Nanoscale Neuronal Connectivity

Ali K. Al-Awami, Johanna Beyer, Hendrik Strobelt, Narayanan Kasthuri,
Jeff W. Lichtman, Hanspeter Pfister, and Markus Hadwiger

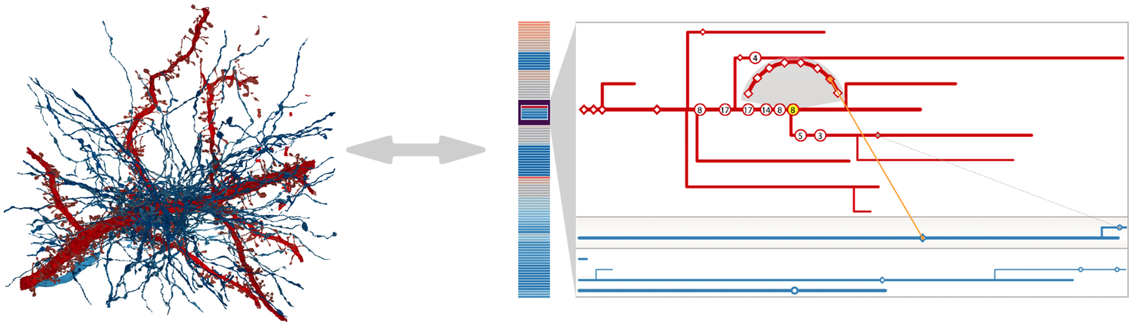


Fig. 1: ***NeuroLines neurite visualization.*** *We abstract the original 3D structure and topology of neurites segmented in nanoscale brain tissue data into a 2D subway map visualization that preserves topology and relative distances. Left: Volume rendering of a dendrite (red) and connected axons (blue). Right: NeuroLines abstraction of the same data, represented as subway lines to more clearly show branches, clusters of adjacent synapses, individual synapses, and the actual connections (shown on demand).*

Abstract—We present *NeuroLines*, a novel visualization technique designed for scalable detailed analysis of neuronal connectivity at the nanoscale level. The topology of 3D brain tissue data is abstracted into a multi-scale, relative distance-preserving subway map visualization that allows domain scientists to conduct an interactive analysis of neurons and their connectivity. Nanoscale connectomics aims at reverse-engineering the wiring of the brain. Reconstructing and analyzing the detailed connectivity of neurons and neurites (axons, dendrites) will be crucial for understanding the brain and its development and diseases. However, the enormous scale and complexity of nanoscale neuronal connectivity pose big challenges to existing visualization techniques in terms of scalability. NeuroLines offers a scalable visualization framework that can interactively render thousands of neurites, and that supports the detailed analysis of neuronal structures and their connectivity. We describe and analyze the design of NeuroLines based on two real-world use-cases of our collaborators in developmental neuroscience, and investigate its scalability to large-scale neuronal connectivity data.

Index Terms—Connectomics, Neuroscience, Data Abstraction, Multi-Trees, Focus+Context.

1 Introduction

Neuroscientists in the field of connectomics seek to reconstruct the full anatomical and functional connectivity of the brain at the resolution of individual connections (*synapses*) between nerve cells (*neurons*). Determining—and ultimately decoding—this wiring diagram, called the *connectome*, is one of the main scientific endeavors of the 21st century and will allow scientists to better understand how the brain develops and functions, and how memories are formed and recalled. However, even a single cubic millimeter of a mouse brain already consists of around 100,000 neurons and 700 million synapses, making brain connectivity exceptionally difficult to analyze and understand.

Only recent advances in connectomics have made it possible to acquire data at the speed and quality necessary to be able to reconstruct the brain's connectivity at the level of individual synapses. Until now, most of the effort has focused on developing novel methods for high-throughput and high-resolution image acquisition [30], data registration [12], segmentation [21, 24], synapse identification, and the reconstruction of connectivity [9, 43]. The next logical step—the detailed analysis of reconstructed neurons, using knowledge of their morphology and structure as well as connectivity—is still a cumbersome, and mostly manual, process. This is further complicated by a lack of sufficiently powerful visualization and analysis tools, and by the inherent complexity and huge scale of connectomics data. Most previous methods for visualization and analysis in connectomics focus either on visualizing a completely abstract connectivity graph, or on rendering the raw electron microscopy (EM) data. A common problem for visualization is the complex structure of EM data. For example, a single neuron in the mouse cortex makes up to ten thousand synapses with other neurons, which easily leads to cluttered visualizations. Furthermore, in order to gain insight, the synaptic connectivity between neural structures and the anatomical information present in the raw data should be linked together, to enable scientists to jointly analyze these two aspects and their relationship. Therefore, it is necessary to design and develop new scalable visualization and analysis techniques that enable scientists to efficiently form and investigate new hypotheses about the interplay between neuron form, connectivity, and function.

In this paper, we present *NeuroLines*, a novel multi-scale visualization technique to analyze neurites (i.e., axons and dendrites, which are structural sub-parts of neurons), and their connections. We use the underlying anatomical tree structure of neurites to create an intuitive visualization. Therefore, our first contribution is an abstraction for visualizing neurites in an uncluttered 2D representation. Each neurite is represented as a tree structure based on its real, but adaptively simplified, anatomy, and its branches. The goal of our visualization is to present neurites in an uncluttered fashion, while still preserving the topological structure and connectivity information. Fig. 1 shows the

- *Ali K. Al-Awami and Markus Hadwiger are with King Abdullah University of Science and Technology (KAUST).*
  E-mail: {ali.awami, markus.hadwiger}@kaust.edu.sa.
- *Johanna Beyer, Hendrik Strobelt and Hanspeter Pfister are with the School of Engineering and Applied Sciences at Harvard University.*
  E-mail: {jbeyer, hstrobelt, pfister}@seas.harvard.edu.
- *Narayanan Kasthuri and Jeff W. Lichtman are with the Center for Brain Science at Harvard University.*
  E-mail: bobby.kasthuri@gmail.com, jeff@mcb.harvard.edu.

Manuscript received 31 Mar. 2014; accepted 1 Aug. 2014; date of publication xx xxx 2014; date of current version xx xxx 2014.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

original 3D structure of a dendrite with several connected axons, and the corresponding representation in NeuroLines. Our second contribution is a multi-scale visualization and navigation scheme that makes our approach scalable to thousands of neurites by automatically computing the correct level of abstraction for the current view. Users can zoom in on regions while still maintaining a contextual overview of nearby neurites. Our third contribution is the NeuroLines application, based on the topology-preserving visual abstraction of neurites to support the analysis of neurite connectivity. It is integrated into ConnectomeExplorer [5], a visualization and visual analysis framework for petascale connectomics data. Finally, our fourth contribution is a demonstration of the utility of NeuroLines, based on two case studies performed by domain experts on real-world connectomics data.

2 Related Work

Connectomics. There are several excellent introductions to the main research challenges in connectomics [30, 40]. In computer science, connectomics has stimulated a lot of research in image processing, vision, and visualization. Most software for connectomics focuses on manual annotation [2, 19, 38] and (semi-)automatic segmentation [1, 21, 24] of neuronal structures as well as proof-reading automatic segmentation results [18, 37]. However, all of these approaches do not support advanced exploration or visual analysis of features.

Visualization for connectomics. There are two main categories for visualization in connectomics: (1) displaying the original large-scale microscopy data, and (2) visualizing higher-level connectivity information [36]. Margulies et al. [32] give an overview of different frameworks for visualizing the human connectome. Hadwiger et al. [17] present a system for volume exploration of petavoxel EM data, which was later extended to handle segmented neurites [6]. Several applications have proposed interactive or visual queries to explore these typically very large data sets [10, 31, 41]. None of them, however, focus on exploring the connectivity between neurites at the level of individual synapses. More recently, we have presented ConnectomeExplorer [5], which supports fully dynamic visual queries, volume visualization of EM and segmentation data, as well as labeled meta-data such as synapse locations. However, in contrast to NeuroLines, ConnectomeExplorer does not focus on detailed neural connectivity analysis and does not offer visual abstractions of neurites and their synapses.

Neuronal connectivity visualization. The analysis and visualization of the intricate connectivity of brain networks typically focuses on either the regional or the cell level. On the regional level, Irimia et al. [20] use connectograms or radial network layouts to show the connectivity between regions in the human cortex. Jianu et al. [22] project 3D tractography data of white matter fibers onto 2D planes to produce 2D neural maps of fiber tracts. Li et al. [29] propose a toolkit for visual analysis of brain networks based on DTI data. Connected brain regions are displayed in a 3D graph structure based on the region's spatial location in the brain. Similarly, the Connectome Viewer Toolkit [15] supports the analysis of macroscopic neuronal structures and brain region connectivity. All of these techniques focus on the high-level connectivity of entire brain regions and do not operate on the level of individual synapses, which is required for nanoscale connectomics. On a cell level, the Viking Viewer [2] displays an abstract connectivity graph, representing each neuron as a single node. ConnectomeExplorer [5] also displays connectivity information in an abstract graph, with one node representing a single neurite. More recently, neuroMap [43] has employed circuit wiring diagrams to represent all possible connections of neurons. All of these methods display neuron connectivity as an abstract network or graph. While this simplifies the visual representation, it also removes the inherent anatomical and topological information of neurites, such as their branches, branch positions, or synapse distribution, and does not support the analysis of individual synapses.

Trees and multi-trees. Neurons and neurites can be represented as topological trees. Consequently, a collection of neurites can be thought of as multi-trees or a forest. Synapses between these neurites result in trees that are connected on a synaptic level. A good introduction to trees is given in surveys on visualization of trees [39] and multi-trees [16]. Fitch and Margoliash [14] construct phylogenetic trees that highlight the mutation distance between two nodes. TreeJuxtaposer [34] focuses on the structural comparison of large trees. It uses a similarity measure to compute the best corresponding nodes between trees and guarantees the visibility of these nodes on screen. Bremm et al. [8] compare multiple phylogenetic trees based on global as well as local tree structure. Tree comparison approaches are very useful to highlight differences between individual trees. Our primary focus in NeuroLines, however, is the analysis of how individual trees (i.e., neurites) are *connected* to each other, and to identify appropriate attributes that can be used for automatic neurite comparison in the future. EVEVis [33] visualizes large evolutionary tree data. It uses a multi-scale method that transitions from a high-level stack graph visualization to a node-link tree layout for showing individual cells. In the medical field, tree structures often have a direct spatial correspondence. CPR (curved planar reformation) [23] allows tubular structures such as vessels to be displayed in a 2D visualization with minimal loss of information. More recently, Borkin et al. [7] proposed a 2D visualization of artery trees for the diagnosis of heart disease. Both methods focus on anatomical structures represented as a single tree, and therefore do not have to deal with multi-trees and connected multi-trees. The metro map metaphor, originally used to show transit lines and connecting stops, can also be used to visualize networks and abstract graphs. Examples include project plans [44] or visualizing trains of thought [35].

Multi-scale navigation. Hierarchical navigation metaphors and focus-and-context techniques [42] are very useful for exploring a large number of entities or data points. NeuroLines uses the "search, show context, expand on demand" metaphor for exploring large graphs [46], and additionally employs a multi-tier focus-and-context visualization to navigate neurites at different levels of abstraction and also allows multi-criteria sorting. Multi-tier focus-and-context techniques have also been used for visualizing large heat maps of genomics data [28]. Telea [45] uses extended table lenses and treemaps to display large tabular data. The system supports multi-column sorting and visually enhances the sorting result to allow easily distinguishing different clusters when the number of sorted elements is high.

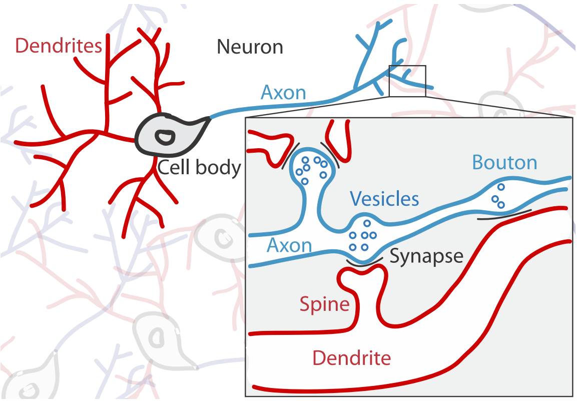


Fig. 2: ***Neuron and synapse structure.*** *Neurons receive input through dendrites, and transmit signals over their axon and the synapses that it makes with other neurons' dendrites. Vesicles carry neurotransmitters on the pre-synaptic side, at so-called axon terminals (boutons).*

3 Biological Background

This section introduces the biological background and terminology, and the data acquisition and processing workflow of our collaborators.

3.1 Neuroscience Terminology

The mammalian brain consists of hundreds of billions of interconnected nerve cells—the *neurons*. Each neuron processes and transmits information, mostly as electrical signals, by forming synaptic connections with other neurons. A single neuron typically consists of a cell body, several dendrites, and one axon (see Fig. 2). A *dendrite* is a tree-like branching structure that receives signals from neighboring cells, while an *axon* is a long and narrow tubular structure that transmits signals away from the cell body towards other neurons. Axons and

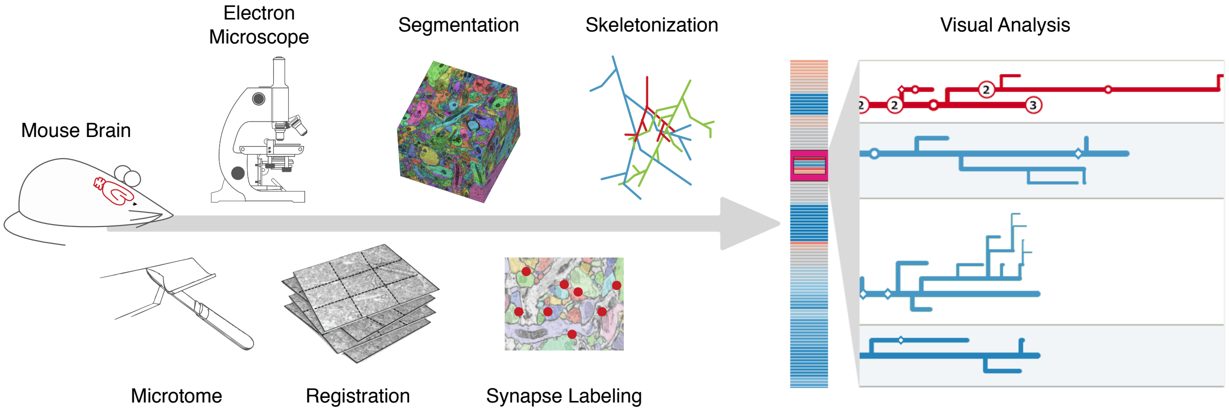


Fig. 3: ***Data processing workflow.*** *After slicing and imaging a block of brain tissue, registration, segmentation, and synapse labeling are performed. The segmented neurites are then skeletonized, forming the basis for subsequent interactive visualization and analysis in NeuroLines.*

dendrites are collectively called *neurites*. A *synapse* consists of a presynaptic terminal (a bouton) on the side of the axon that releases neurotransmitters when activated, a post-synaptic terminal at the side of the dendrite, and the synaptic cleft between dendrite and axon.

A synapse can either be *excitatory* or *inhibitory*, depending on whether a spike on the axon increases or decreases the chance of provoking a spike in the receiving dendrite. Synapses have further discriminating features, such as the position of the post-synaptic terminal (i.e., either on a dendrite's shaft or on a small extension called a *dendritic spine*), or the number of *vesicles* (cell organelles containing neurotransmitters) in the bouton. However, little is known about what influences or causes the variability of these attributes, and how they impact the function of a neuronal network. Scientists are interested in trends and correlations, and in looking at individual neuronal structures, their synapses, and attributes. For example, they want to look at detailed synapse characteristics to find patterns depending on specific axon/dendrite constellations: If an axon makes several connections to the same dendrite, do all their shared synapses look the same? Do synaptic pathways follow certain excitatory/inhbitory patterns?

3.2 Neuroscience Workflow

The data acquisition and processing workflow of our collaborators consists of several steps (see Fig. 3): Starting with a solidified block of brain tissue (a tiny sample of a mouse or rat brain), they use an ultramicrotome to cut it into slices of 25-30 nm thickness, which are then scanned with a scanning electron microscope (SEM) to capture image tiles with a pixel resolution of 3-5 nm. Next, the individual tiles are stitched and registered to form a single 3D volume with slice resolutions of 20,000 to 100,000 pixels, and thousands to tens of thousands of slices. In the next step, our collaborators segment and label this 3D volume, using both manual and automatic segmentation tools [24]. Mojo [25] and Dojo [18] are used for proof-reading automatically generated segmentations. All additional data, such as the locations and properties of synapses, are currently annotated manually and stored as meta data. The necessary processing steps for importing these initial data into NeuroLines for further analysis are explained in Sec. 8.1.1.

Currently our collaborators use several different tools for data analysis. To support their workflow, we have integrated NeuroLines as a plug-in into ConnectomeExplorer [5], which is a system for visualization and visual analysis of large-scale neuroscience data that supports interactive visual queries to dynamically explore data. It supports 3D volume rendering, and offers an abstract node-link diagram to depict neurite connectivity. For detailed statistical analysis of regions of interest in the volume, our collaborators mainly use Matlab.

4 NeuroLines Design

The main idea of our design is to abstract the complex branching and connectivity pattern of neurites into a simplified representation inspired by 2D subway maps. We transform the problem from 3D into 2D to reduce visual clutter, while preserving branching patterns and relative synapse locations, and to facilitate following synaptic chains. Fig. 5 shows an overview of the NeuroLines system.

4.1 Design Considerations

The idea of NeuroLines originated in initial meetings with our collaborators where they voiced their dissatisfaction with the lack of neurite visualization approaches that focus on connectivity instead of on a complete 3D reconstruction of the segmented structures. Our first prototype depicted neurite connectivity as an abstract 2D node-link diagram (Fig. 4 (c)). It was included in the ConnectomeExplorer framework [5]. This view allowed our collaborators for the first time to see the connectivity of their data. However, this approach loses spatial relations and knowledge of the branching morphology. Next, our collaborators wanted a combined visualization of the original 3D data and a simplified 3D graph structure, which they called a "3D subway map." However, after several iterations of such a design (Fig. 4 (b,d)), it became clear that a 3D approach does not scale to the expected data sizes, and that it leads to nonintuitive and cluttered visualizations, ultimately motivating a novel 2D representation. This 2D abstraction simplifies the original 3D structure and removes anatomical details. However, it retains the most important features for subsequent analysis: topological structure, connectivity information, and synapse sequence along a neurite. Additionally, we always allow users to go back to the original 3D volume view from any point in the 2D representation.

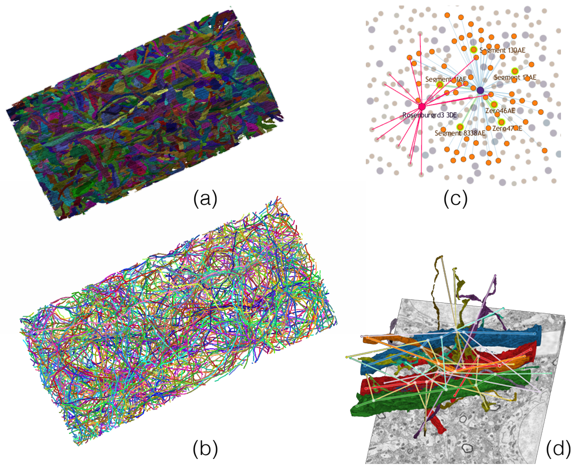
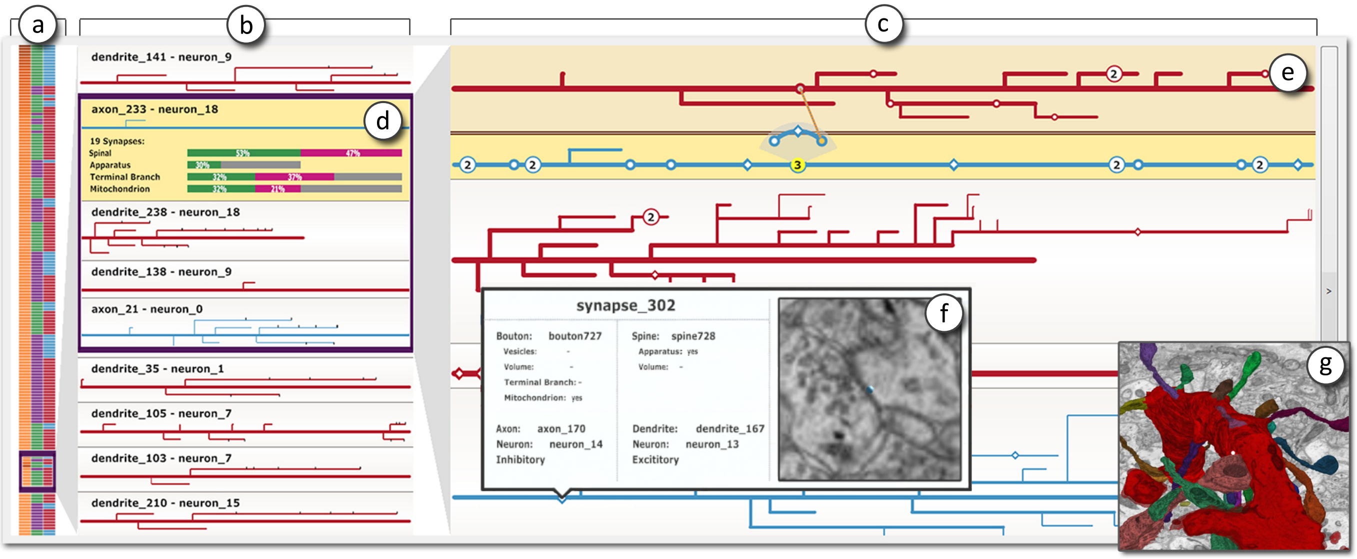


Fig. 4: ***NeuroLines design prototypes.*** *a) 3D volume rendering of a cylindrical region of interest of the segmented data; b) First visualization approach in 3D for directly displaying neurite skeletons; c) Abstract 2D graph visualization showing the connectivity between neurites, without spatial information; d) First 3D subway map prototype.*



19 Synapses:
Spinal 53% 47%
Apparatus 30%
Terminal Branch 32% 37%
Mitochondrion 32% 21%

synapse_302

Bouton: bouton727
Vesicles: -
Volume: -
Terminal Branch: -
Mitochondrion: yes

Spine: spine728
Apparatus: yes
Volume: -

Axon: axon_170
Neuron: neuron_14
Inhibitory

Dendrite: dendrite_167
Neuron: neuron_13
Excititory

Fig. 5: ***NeuroLines overview.*** *We visualize neurites using a multi-scale approach with three tiers of linked views (a,b,c), which provide overview as well as details. The* navigation bar (a) *shows individual neurites as lines color-coded according to selected neurite attributes and sorting criteria. The* neurite overview (b) *allows inspecting all neurites at a medium level of abstraction. The* workspace view (c) *allows the inspection of neurites at a detailed level of abstraction. The* neurite analysis (d) *shows statistics for a selected neurite.* Pinning (e) *a neurite to the workspace allows keeping a specific neurite in focus while exploring others. The* synapse analysis (f) *shows synapse details and its neighborhood in the original EM volume.* (g) *NeuroLines is also coupled with a 3D volume renderer for additional visualization and exploration in 3D.*

4.2 Task Analysis

This section discusses our overall domain goals and present the corresponding detailed task analysis. These analyses have guided most design aspects of NeuroLines.

4.2.1 Domain Goals

The main objective of our collaborators is to quickly form, test, and accept or reject new hypotheses regarding neuron connectivity. This includes discovering patterns, exploring a region of interest or subset of the data, and to quickly identify segmentation or labeling errors. NeuroLines supports the overall goals of neuroscientists to **(a)** explore and identify patterns in **synaptic connections**; **(b)** explore and identify patterns in **branching structures**; and **(c)** explore **synaptic pathways**.

An example of (a) is the analysis of multiple-hit axons and the distribution of their synapse locations on the post-synaptic dendrite. Scientists want to explore the connectivity pattern of where a single axon connects to the same dendrite multiple times, and see the locations of these synapses on the dendrite—and in relation to other synapses of the dendrite. An example of (b) is the analysis of dendritic synapse strength with respect to their distance from the cell body. Synapses further away from the cell body are assumed to be bigger, so that the signal they send towards the cell body is stronger. Our collaborators also want to analyze the effect of branching on synapse strength. An example of (c) is the identification of recurring connectivity motifs. Our collaborators look for certain recurring connectivity patterns in the data, such as "inhibitory neuron → excitatory neuron → excitatory neuron → inhibitory neuron". In order to do this, they have to follow synaptic pathways and identify neurons as excitatory or inhibitory.

4.2.2 Domain-Driven Tasks

We now map our high-level domain goals to analysis tasks that need to be supported by NeuroLines. We identified these tasks over several months of meetings with our domain scientists through semi-structured interviews and informal feedback sessions. Initial meetings were held in bigger groups of several neuroscientists and scientific staff, while detailed discussions were done with individual scientists. We have identified the following main tasks:

T1–Selecting a neurite subset. In addition to a typical "overview first, details on demand" visual exploration scenario, our collaborators want the possibility to start their exploratory process with a specific subset of the data. For example, it is interesting to consider only neurites inside a specific spatial region of interest, or start with only excitatory axons that were sorted according to their number of synapses.

T2–Single-neurite analysis. Typically, after selecting a structure of interest, i.e., a neurite, it is explored in more detail before continuing with further analysis of connected or nearby structures. This detailed analysis includes examining the neurite's attributes such as branching complexity and length, as well as synapse distribution and statistics over all synapses of a neurite.

T3–Multi-neurite analysis. When analyzing neurites, it is important to be able to determine relationships with other neurites. Looking at different neurites concurrently and in the same view allows users to quickly extract patterns like branching structure, connectivity, function or length, and to compare individual neurites to each other.

T4–Synapse analysis. Synapses are the basic elements that create neuronal connections and pathways. Before analyzing the connectivity on a larger scale, individual synapses have to be analyzed and classified. For example, the combination of different synapse attributes is correlated with the strength of a synapse. It is crucial for the scientists to be able to examine individual synapse attributes, to look at the original synapse location, and to navigate to this location in a 3D view.

T5–Connectivity analysis. Following synaptic connections from one neurite to another is crucial for further understanding of the underlying data. For example, being able to explore a specific axon and all its connections to a specific dendrite allows analyzing multiple-hit axons and extracting related properties, such as the synapse distribution along a neurite, or the strength of synapses on all first-level branches (i.e., the branches directly off of the trunk of the neurite).

The mapping of domain goals to tasks is as follows: exploring synaptic connections (a) is supported by tasks T1, T4, and T5; exploring branching structures (b) is supported by T1, T2, and T3; and exploring synaptic pathways (c) is supported by T1, T3, T4, T5.

4.3 Scalability Challenges

Our collaborators are constantly working on increasing the size (i.e., the physical extent as well as the resolution) of their data. Therefore, one of our main design goals was to develop a scalable visualization that enables hierarchical navigation through a large set of neurites. To test the scalability of our system, we have implemented a parameterized neuron simulator (Sec. 8.1.2) that allows us to create synthetic data for stress testing each of the following scalability challenges:

S1–Many neurons. The current data set our collaborators have scanned is too "small" in physical extent to figure out neuron/neurite relationships (i.e., which neuron a specific neurite belongs to). The high resolution of EM results in teravoxel datasets for a tissue block of only several cubic micrometers. Axons, however, can extend over a distance of milli- and centimeters. With the rapidly increasing amount of scanned data, however, we will have access to hundreds if not thousands of segmented neurons over the next few years.

S2–Many neurites. As the number of segmented neurons increases, the number of neurites will increase even more (by 1-100x). It will become difficult to navigate through large lists of neurites, and find structures of interest while still seeing their immediate context.

S3–Many branches. A neurite can consist of dozens to hundreds of branches, resulting in very big individual neurites with a potentially confusing branching structure.

S4–Many synapses. A neurite can have hundreds of labeled synapses, producing a lot of clutter if not reduced by the visualization.

S5–Many connections between neurites. As the number of labeled synapses increases, showing visual links between all connected neurites becomes infeasible. Moreover, since a neurite can have hundreds of synapses, they might connect to neurites that are currently outside the subset of the data that is visible on screen.

5 Visual Elements

This section gives a high-level overview of the major visual elements of NeuroLines, from the main view comprising a multi-scale view with three different tiers (Fig. 5 (a,b,c)), to the different abstraction levels of neurites (Fig. 6). We use neurites as our main visual representation instead of neurons because (a) this is the level of detail for connectivity analysis (i.e., axons connect to dendrites), and (b) currently many of the neuron-neurite relationships are still unknown in our data.

5.1 Multi-Scale, Three-Tier Main View

Fig. 5 depicts the main view of NeuroLines, which consists of a multi-scale, three-tier focus-and-context neurite visualization. It comprises the following three tiers, which are arranged from left to right:

1. The *navigation bar* for a high-level overview of the neurite *working set*, and for dynamic sorting and navigation.
2. The *neurite overview* to get an overview of a subset of neurites, and for inspecting high-level neurite information.
3. The *workspace view* for inspecting individual neurites, branching patterns, synapses, and synaptic chains in detail.

These views allow simultaneously navigating the data at different abstraction levels, from overview to detail. Using a three-tier focus-and-context scheme enables the exploration and analysis of multiple neurites at the same time, at different scales (T3–multi-neurite analysis, S2–scalable to many neurites).

The zoom levels in all views are set automatically, but can be adjusted dynamically by the user. Changing the zoom level adjusts all contained elements, and updates the amount of branch collapsing, synapse clustering, and text overlays accordingly. All three views are linked by sliding focus windows that allow drilling down into the data from left to right (Fig. 5 (a,b,c)).

We use the concept of a *working set* to denote the current total set of neurons and neurites that are of interest to the user. The initial working set is the entire data set, but the embedded visual query language [5] can be used to dynamically restrict the working set to the objects resulting from an interactive query. In this way, NeuroLines supports the "search, show context, expand on demand" interaction metaphor [46].

5.1.1 Navigation Bar

This view (Fig. 5 (a)) represents neurites as horizontal, color-coded lines. Users can navigate all neurites vertically, visually identify patterns, and navigate and drill down into areas of interest.

The design and color-coding of this view is inspired by heatmaps where neurites are color-coded either according to user-chosen neurite attributes (e.g., function, number of synapses), or by the color that was used for the initial segmentation in the EM volume (Fig. 7). For automatic color-coding, we use sequential and qualitative color schemes. The colors that were chosen manually by the scientists during segmentation are arbitrary. However, the scientists often identify individual structures via certain colors, and therefore want to be able to use the same colors in NeuroLines. The navigation bar supports multi-criteria sorting of neurites, which is explained in more detail in Sec. 6.1.

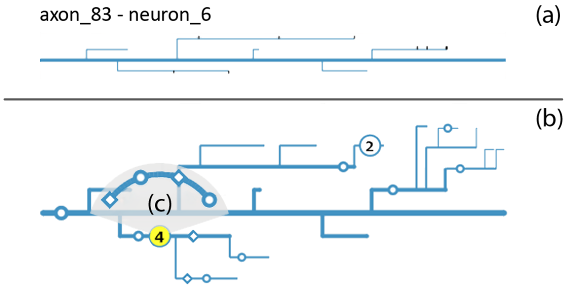


Fig. 6: ***Neurite abstraction.*** *We employ different abstraction levels to visualize neurites at different levels of detail. (a) Medium-level abstraction showing an overview with collapsed branches. (b) Detailed view (low-level abstraction) with individual synapses shown as diamonds (spinal) or circles (non-spinal). (c) Synapses overlapping in screen space are automatically clustered. Clusters show the number of contained synapses and can be fanned out to show them in sequence.*

5.1.2 Neurite Overview

This view (Fig. 5 (b)) depicts neurites at a medium level of detail in order to give an overview of neurites, i.e., it shows neurites without the full detail of their branches or individual synapses. Each neurite in this view is shown with its textual name. Additionally, neurites can be selected to see summary statistics of their attributes (Fig. 8), which is explained in Sec. 7.1.

5.1.3 Workspace View

This view (Fig. 5 (c)) acts as the main *workspace* in NeuroLines, for detailed analysis of neurites, their branching patterns, synapses and connections, and for comparing multiple neurites.

Each neurite is represented as a horizontal tree that depicts the neurite's complete branching structure and all of its synapses, depending on the zoom level. The view can be zoomed and panned both vertically and horizontally. By default, each neurite is scaled horizontally such that it fills up the available space, while preserving the relative distances of branches and synapses within the neurite. When zooming out, the neurite tree visualization is gradually simplified by collapsing branches and merging individual synapses into synapse clusters based on the current resolution level and zoom factor.

5.1.4 On-Demand Electron Microscopy Views

A main requirement of our collaborators is that they want to be able to go back to the original 3D data to explore the detailed anatomy of the area around a synapse. Therefore, we have integrated on-demand 2D and 3D volume views of the original and segmented EM data that can be activated by clicking on a synapse (Fig. 5 (f,g)). We display a small z-aligned 2D slice view that is centered at the x,y,z location of the synapse in the original EM volume showing the immediate synapse neighborhood (T4–single synapse analysis). Furthermore, we can automatically navigate to the synapse in a 3D volume rendering view that shows the synapse and the segmented axon and dendrite it connects, and interactively explore the entire EM volume in the 3D view.

5.2 Neurite Abstraction Levels

To support exploration and analysis at different levels of detail, NeuroLines employs neurite visualizations with different abstraction levels. The highest abstraction level is used in the navigation bar, where each neurite is represented as a single horizontal, color-coded line. Fig. 6 shows the medium and low-level abstraction levels that we provide.

The medium-level neurite abstraction shows an overview of the neurite in which all but the most prominent branches are collapsed. This abstraction is used for visualizing neurites in the *neurite overview* (Fig. 5 (b)). Additionally, a neurite can be augmented by statistical information about its synapses (Fig. 5 (d)).

The low-level abstraction of a neurite retains most details and is used for visualizing neurites in the *workspace view* (Fig. 5 (c)). The specific amount of detail shown depends on the used zoom level.

5.2.1 Computing Abstracted Neurites

To draw an abstracted neurite, we map its 3D skeleton structure to a simplified, but topologically correct, 2D representation inspired by subway maps. We preserve all relative distance relations within a neurite (i.e., distances between synapses/branches, lengths of branches), but straighten branches to obtain a visualization with clear and straight lines (T2–single neurite analysis). We offset branches at right angles, allowing a direct comparison of horizontal positions between parallel branches and synapses. Neurites are scaled horizontally to either maximize use of available screen space, or scaled in relation to a selected neurite (Fig. 3, right). We use relative instead of absolute scaling because individual neurites can differ in length by several orders of magnitude. This makes an absolute scale useless in most cases.

Branching. We use a greedy approach to draw the branching structure of neurites: After drawing the trunk, we iteratively add branches from right to left, aiming for a compact visual representation. We avoid screen space intersections of different branches by alternating the vertical position at which branches are added (either at the top or at the bottom). If this does not resolve conflicts, we increase the distance of branches from the trunk. The current branching level is visually encoded via thickness and vertical distance to neighboring branches.

To prevent neurites with many branches from dominating screen space, we automatically collapse branches vertically if the height of a neurite exceeds the allowed maximum height for the current resolution level and zoom factor (S3–scalable to many branches). More specifically, the allowed maximum vertical distance between branches is set differently for each resolution level (i.e., overview and workspace view) and is automatically decreased for each additional branching level. If we detect that a branch would exceed the allowed vertical distance, it is automatically collapsed starting with its sub-branches.

Synapses. Synapses are not part of the initial neurite segmentation skeleton. Therefore, we project their labeled 3D position onto the corresponding skeleton element and display them as small nodes on the neurites. This implies that each synapse is displayed twice: once on the axon, and once on the dendrite. We encode the post-synaptic morphology by displaying spinal synapses as diamonds and non-spinal synapses on a dendrite's shaft as circles. For scalability to a large number of synapses (S4–scalable to many synapses) we use mean-shift clustering to group synapses that overlap in screen space into clusters and display each cluster as a single node, displaying the number of contained synapses. Clustering is based on the horizontal position of synapses and is done for each branch separately. Synapses retain their spatial order inside a cluster and can be explored individually by fanning out the cluster's elements upon selection.

Synapse links. To reduce the amount of visual clutter, we draw visual links (i.e., synaptic connections) between neurites only on-demand, when hovering over a synapse (S5–scalable to many connections between neurites). Displaying stubs instead of lines [27] would be a viable alternative to reduce clutter while still indicating connections. However, in our case a synapse node on a neurite already indicates the presence of a connection. In addition to showing the visual link of the selected (i.e., hovered) synapse, we also highlight all synapses between the same two structures. This allows the user to quickly identify not only the number of shared connections between two structures, but also to examine each synapse in more detail.

6 Interaction

In this section we focus on the interaction features of NeuroLines. The overview in Fig. 5 depicts many of the interaction possibilities that we provide, including multi-criteria sorting (a), the concept of a workspace (c), pinning neurites of interest (e), selection of individual synapses (e,f), and joint 2D/3D data exploration (f,g).

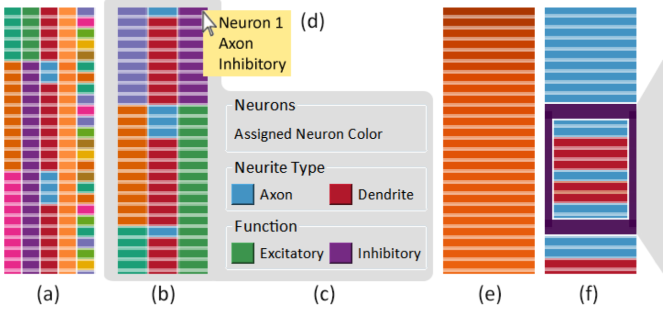


Fig. 7: ***Neurite sorting and high-level navigation.*** *The current sorting of neurites is depicted in the navigation bar by color-coding the sorting criteria according to neurite attributes. (a) Multi-critera sorting order from left to right: neuron id, function (excitatory/inhibitory), type (axon/dendrite), no. synapses, assigned neurite color. (b,c) Multi-critera sorting and color map (left to right: neuron id, type, function). (d) Tool tip for a single neurite showing its attributes. (e) Neurites can also be color-coded with a single attribute (no. synapses) regardless of the sorting criteria. (f) The navigation bar also displays a slidable focus window, which is linked to the neurite overview for navigation.*

6.1 Sorting and Filtering

To deal with a large set of neurons and neurites (S1, S2–scalable to many neurons and neurites), we support filtering and sorting operations to dynamically find and define subsets of the data.

Filtering. For powerful filtering operations, we use the concept of a *working set*, which is determined using the integrated dynamic query language [5]. By default, the working set is the entire data set. However, dynamic queries allow the user the on-the-fly specification of a set of objects of interest, such as a specific set of neurites or synapses, that should be analyzed together. This helps narrowing down the analysis to the subset of the data that is needed for a specific analysis task (T1–select neurite subset). For example, to examine a specific dendrite, our collaborators specify queries to extract this dendrite, all its connected axons, and all dendrites that connect to these axons. This significantly reduces the amount of data that needs to be displayed.

Multi-criteria sorting. The working set can be sorted according to multiple criteria or user-specified categorical and quantitative attributes (e.g., neurite function, number of synapses), to facilitate the discovery of high-level patterns in the data. The current sorting is displayed in the navigation bar and uses a heatmap approach to display all sorting criteria in a condensed form where each sorting attribute is shown as its own vertical column (Fig. 7). The individual columns are arranged from left to right depending on their sorting sequence, with left being the first sorting attribute. Sorting attributes and their sequence can be defined in a GUI widget which also shows the color map for each attribute. Furthermore, we allow hovering over individual entries in the navigation bar to inspect their values in more detail.

6.2 Workspace, Pinning, and Pivoting

To allow users to track their progress and to compare neurites, we use the concept of a *workspace*, where neurites can be saved and stored even when the user goes on to explore different parts of the data set (T3–multi-neurite analysis, S2–scalable to many neurites).

Pinning. By *pinning* a neurite to the workspace (Fig. 5 (e)), it is guaranteed that it stays visible even when the remainder of the view changes, thereby allowing comparisons between different neurites.

Pivoting. By selecting individual neurites in the workspace, we can sort and scale neurites based on this item (i.e., pivot element), and easily explore a neurite's neighbors (T5–connectivity analysis). We support re-ordering the neurites depending on features such as connectivity strength, where neurites with many synapses to the selected neurite are arranged more closely than neurites with fewer synapses.

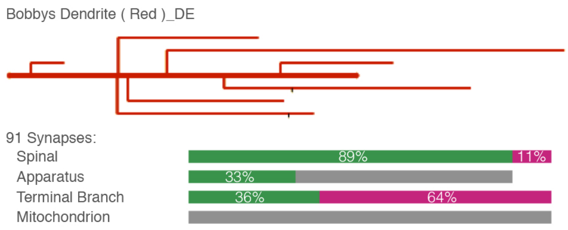


Fig. 8: ***Neurite overview and analysis.*** *This view combines a medium-level abstraction of a neurite with detailed statistics over all of the neurite's synapses, e.g., percentage of spinal vs. non-spinal synapses.*

Scaling to pivot. In the same manner, we can scale the length of neurites (the horizontal axis) according to the length of the pivot element, to better compare the size, branching pattern, and synapse locations between multiple neurites (T3–multi-neurite analysis).

6.3 Connectivity Exploration

One of the main goals of NeuroLines is connectivity exploration and the discovery of connectivity patterns. Ultimately, our collaborators want to identify connectivity motifs in their data (e.g., neuronal feedback loops). NeuroLines supports this task by allowing scientists to quickly explore the connectivity between neurites manually, but in the future we also want to integrate (semi-)automatic motif detection.

Once the user hovers over a synapse, it is displayed as a visual link (i.e., connecting line) between the respective synapse locations on both the axon and the dendrite. In addition to highlighting the currently selected synapse and its connecting line, all other synapses between the same two neurites are displayed as visual links for contextual information (Fig. 10 (c)). This allows users to quickly see how many synapses are shared between two neurites and to follow synaptic chains throughout the dataset. Connected structures outside the current viewing window can be fetched and moved next to the originating neurite in a smooth animation. This keeps the current synapse in focus, but allows the user to explore the connected structure at the same time (T5–connectivity analysis, S2, S5–scalable to many neurites and many connections between neurites) . Optionally, the viewport can be moved to the neurite in question in a smooth transition.

7 Analysis Tools

This section focuses on the analysis features of NeuroLines that support scientists in exploring and validating their hypotheses.

7.1 Neurite Analysis

The *neurite analysis view* depicts detailed statistics over all of the neurite's synapses in an easy to read color-coded stack chart (Fig. 8). This view is integrated into the *neurite overview* and shown only on-demand for the currently selected neurite in the workspace, as depicted in Fig. 5 (d). This view allows scientists to see the most important statistics of a neurite at a single glance, allowing them to quickly identify trends and patterns. For example, if the function of a neurite (i.e., excitatory or inhibitory) is unknown, scientists can try to infer the function based on the percentage of spinal synapses.

7.2 Synapse Analysis

To inspect individual synapses in detail, NeuroLines offers a *synapse analysis view* that is displayed on-demand. This is triggered by clicking on a synapse in the workspace view. This view displays all attributes of the synapse, including information about the pre-synaptic bouton and the post-synaptic terminal. Additionally, it offers a 2D slice view and a linked 3D volume view centered around the synapse, which are shown in Figs. 5 (f,g). This allows scientists to confirm synapse attributes and look for additional distinguishing features. The case study in Sec. 9.1 is one example of how synapse analysis can help answer domain-specific questions.

8 Implementation and Evaluation

NeuroLines is implemented as a plug-in in the ConnectomeExplorer framework. It is written in C++ and OpenGL, and uses Qt for basic GUI and window elements. The application runs on a standard Windows PC and requires a recent NVIDIA GPU (Kepler architecture or better) to run the 3D volume renderer. The neuron generator for synthesizing data is implemented in Python.

8.1 Data

In this section, we describe the real-world domain data our scientists have acquired and discuss our neuron generator for simulating large neuronal networks that can be displayed in NeuroLines.

8.1.1 Real-World Domain Data

The data our domain scientists want to analyze and that serves as input to our system consists of a collection of segmented and annotated electron microscopy slices of brain tissue (e.g., the mouse cortex), forming a single large 3D volume. Currently, synapses are labeled manually by the neuroscientists, but in the future synapses will be labeled automatically. In a pre-process, we extract curve skeletons of the segmented neurites using 3D medial axis thinning [26] before converting the extracted skeletons into a forest of trees, each tree representing a single neurite. Additionally, in this step we also deal with incomplete or incorrect data, such as wrong segmentations and labeling. An example of this are "island segmentations," where a segmented structure with the same ID has several unconnected components, which is biologically not possible. However, due to manual segmentation errors or incorrect automatic segmentation these cases can happen, and therefore we represent them as unconnected branches of the same tree.

In the final pre-processing step we add the labeled synapses to the skeletons by finding the two corresponding skeletons (i.e., the axon and the dendrite) and inserting the synapse by orthogonally projecting the synapse location onto to the nearest skeleton element. Then we store the generated skeletons as an XML file. When starting up NeuroLines, we therefore are able to load the original EM and segmentation data with the extracted skeletons and connectivity data.

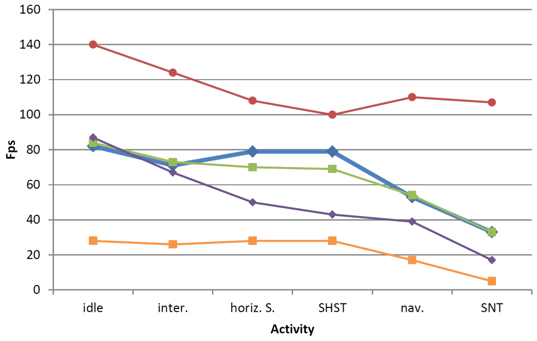
8.1.2 Synthesized Data for Scalability Analysis

After discussions with our domain experts we decided that in order to support future, much larger data, we wanted to be able to evaluate the scalability of NeuroLines by using simulated data. Therefore, over the course of several weeks and in close collaboration with our scientists, we developed a simple parameterized neuron simulator that uses domain knowledge to create neuron skeleton structures and synapses.

Macro structure: neurons and neuron connectivity. Generating neurons is triggered by specifying the mean number of neurons to be created, using a normal distribution to account for variability between different brains. The neuronal network is generated based on the Watts-Strogatz model [47] to create a small-world graph that exhibits local clustering and the formation of hubs, which has been shown to be useful for simulating biological neural networks [4, 11]. Next, we convert the resulting undirected graph into a directed graph by replacing undirected edges with directed edges in a stochastic approach that incorporates our collaborators' knowledge of neuronal connectivity.

Micro structure: neurites and branching patterns. The number of axons and dendrites per neuron as well as the neuron's function is guided by a stochastic process with underlying domain knowledge. Usually a neuron contains one axon (disregarding segmentation errors) and several dendrites, depending on the neuron's function and type. The individual neurite's branching pattern is generated by a context-sensitive, stochastic L-system, that is parameterized based on feedback from the neuroscientists and knowledge of different branching patterns in different neurites (e.g, excitatory axons are longer but exhibit less branching than inhibitory axons). Other systems have already successfully used L-systems to simulate artificial neural structures [3, 13].

Connectivity: Synapse generation. Synapses are generated based on several factors, most importantly neuron function, the number of simulated connected neurons, the size and branching pattern of the current neurite, and a user-set parameter to specify synapse density.



	Dataset	Number of Neurites	Total Branches	Total Synapses
◆	Dataset 1	50,000	695,000	9,000
■	Dataset 2	9,730	134,274	27,244
■	Dataset 3	2,020	424,806	15,716
◆	Dataset 4	1,040	14,664	23,816
●	Real Dataset	646	1,421	776

Fig. 9: ***Performance and scalability evaluation.*** *We have measured frame rates for different data sets and different kinds of user interactions. Data sets 1-4 were generated for testing our system against the scalability challenges (DS1: S1,S2–many neurons, many neurites; DS2: S5–many connections between neurites; DS3: S3–many branches; DS4: S4–many synapses per neurite). Frame rates are for user* idle*, workspace interaction (*inter.*), horizontal scaling (*horiz. S.*), stress horizontal scaling test (*SHST*), multi-tier navigation (*nav.*), and stress navigation test (*SNT*). Standard interaction is always interactive (28-140 fps). During multi-tier navigation, frame rates are lower (17-110 fps) due to the dynamic layout computation and synapse clustering. However, even during very fast multi-tier navigation (SNT) frame rates are still interactive (5-107 fps).*

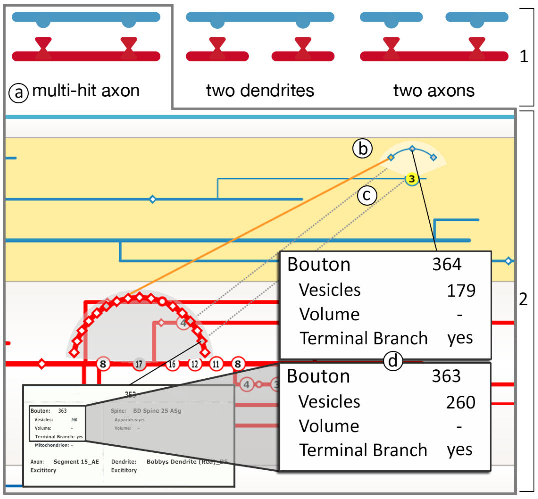
8.2 Scalability Evaluation

We have evaluated the scalability of our system by using synthetic data with different numbers of generated neurons, neurites, branches, and synapses. Generating a data set with thousands of neurites takes several seconds. The rendering of NeuroLines remains interactive, even with more than 50,000 neurites, because we only draw elements inside the viewing window. The high-level navigation bar uses a mipmap-based approach for rendering, reducing the resolution of the data before drawing, when necessary. Detailed frame rates are displayed in Fig. 9. Feedback from the scientists tells us that this number is sufficient, because for larger data they will always filter the data to a subset of interest, using the query algebra, or start with a single object of interest and continue from there. This filtering step is crucial for system usability because it ensures that user interaction maintains effective by limiting the amount of data that needs to be displayed.

We tested branching scalability by generating neurites with up to 1,000 branches. Automatic collapsing and expanding of branches works fine up to ten levels of sub-branching. If that number is exceeded, sub-branches are often collapsed in the detail view which led users to manually expand sub-branches for exploring them in highest detail. In the future, we want to allow detaching sub-branches of interest from the main neurite, to explore them in more detail without cluttering the view with unimportant branches of the same neurite. The maximum number of synapses is currently only limited by the clustering mechanism. If more than 25 synapses are combined into a single cluster, synapses within this cluster should again be partitioned into sub-clusters to avoid visual clutter.

9 Case Studies

We demonstrate the utility of NeuroLines based on two evaluation cases taken from user sessions of our collaborating neuroscientists where they wanted to explore specific research questions. During the entire development and evaluation phase we regularly held meetings with junior and senior level neuroscientists and scientific staff. Two of them are also co-authors of this paper. Both evaluation cases presented here were performed by a developmental neuroscientist with several years of experience in connectomics research and are typical examples of exploratory data analysis, where the scientists adjust and modify their original hypotheses as they advance in their analysis.



Bouton 364
Vesicles 179
Volume -
Terminal Branch yes

Bouton 363
Vesicles 260
Volume -
Terminal Branch yes

Bouton: 363
Vesicles: 260
Volume: -
Terminal Branch: yes
Mitochondrion: -

Spine: BD Spine 25 ASg
Apparatus:yes
Volume: -

Axon: Segment 15_AE
Excititory

Dendrite: Bobbys Dendrite (Red)_DE
Excititory

Fig. 10: ***Case study 1: Synapse variations.*** *Our collaborators focused on multiple-hit axons (a) as neurite connectivity pattern. Subsequent exploration of the red dendrite D1 and one of its synapses (b) shows that it makes three synapses with the blue axon, displayed as visual links between the neurites (c). Looking at the detailed synapse attributes (d) allowed our collaborators to form initial hypotheses about the connectivity of multiple-hit axons.*

The main data set of our primary collaborators used in these cases is an electron microscopy volume of roughly one teravoxel in size (8-bit voxels; 955 GB). The binary segmentation volume, from which we computed the neurite skeletons, was given in a volume of half the resolution in x and y (24-bit voxels; 716 GB), and contains roughly 4,000 segmented objects (i.e., axons, dendrites, and individually labeled dendritic spines). In addition to this, the scientists have labeled 943 synapses with roughly a dozen detailed attributes such as the number of vesicles at the synapse or the spine/shaft location.

9.1 Case Study 1: Relating Variations in Synapse Structure to Neuron Connectivity

This case study was driven by a research question one of our collaborators is working on: "How much of the variance in the structure of synapses can be explained by the connectivity of neurons?" Neuronal connectivity can be thought of in two different ways: a) *Who* does a neuron speak to (i.e., to which other neurons does it connect)? b) *How loud* does it speak (i.e., how often does a synapse fire, and how strong is its electrical signal)? With today's high-resolution EM data sets it is possible to look at both of these attributes at the same time. While the first attribute can be analyzed by looking at the topological connectivity graph, the second part (i.e., how loud a neuron speaks) is not as easy to evaluate. Several attributes influence the strength of neural connections, such as the number of vesicles near a synapse, the area or length over which two neurites touch, whether a neurite is excitatory or inhibitory, as well as the neurite's circumference and the spacing of synapses along a neurite. In this case, the neuroscientist analyzed the variance of specific synapse attributes of different connectivity patterns or motifs. He compared synapse attributes of multiple-hit axons (e.g., axon A that makes several synapses with the same dendrite B) to attributes of single-hit axons (e.g., different axons that connect to dendrite B) to attributes of all non-multiple-hit synapses of axon A and/or dendrite B. Fig. 10 shows a screenshot of the on-going analysis of neurite connectivity patterns. This initial exploration led our collaborators to discover new rules in synapse connectivity, which they subsequently statistically analyzed. Here we give a general overview of how the scientist used NeuroLines to reach his particular goal.

First, he explored the entire data set, sorted all neurites depending on neurite type and the number of synapses, to narrow down on a first structure of interest (i.e., dendrite D1). Using a visual query, the data set was reduced to only include dendrite D1, all its connected axons, and all dendrites these axons connect to. Next, the scientist analyzed the detailed connectivity patterns, starting from dendrite D1. An initial analysis of the attributes of all synapses of this dendrite did not reveal any apparent patterns. Therefore, the scientist first identified several multi-hit axons connected to dendrite D1 (Fig. 10), and then analyzed only the synapses between these axons and dendrite D1. Some of the attributes that the scientist looked at were given as scalar values (e.g., spine volume), while for other attributes (e.g.,"closeness" of both neurites around the area of the synapse) the integrated 2D and 3D views of the original EM data were used. This allowed the scientist to further narrow down his analysis process and to slightly adjust and refine his hypothesis. When he was sufficiently sure of his findings he handed the data over to a statistician to conclude the analysis. Our collaborator was able to perform this analysis in NeuroLines within 2–3 hours, after having received two introductory sessions to our system that both took roughly thirty minutes. Individual analysis steps usually took him between several seconds (for quickly rejecting initial hypotheses) up to 30 minutes (for narrowing in on an hypothesis, using the dynamic queries and then methodically working through the resulting neurites to check the validity of his hypothesis).

9.2 Case Study 2: Relating Variance in Synapse Structure to the Branching Structure of Excitatory Dendrites

In this case study, our collaborator evaluated the branching structure of excitatory dendrites and how this branching structure influences the attributes of synapses along the neurite. A neurite usually gets smaller and narrower at every branching level, meaning that the main trunk of a neurite has a diameter that is significantly larger than the diameter of, for example, a sub-branch of a branch of the trunk. The main question of the scientist was how much of the variance in the structure of synapses could be explained by (a) their absolute distance from the cell, and (b) the branching level from the cell. The fundamental scientific question behind this is related to *how* neurons regulate when a cell fires and when not. Does every vote (i.e., every synapse on the dendrite) have the same influence, even though some synapses have a much larger distance to the cell body than others?

To analyze synapse attributes in relation to the branching pattern of excitatory neurites, our collaborator started by identifying excitatory neurites. When the function of a neurite is unknown, analyzing the number of spine/shaft synapses gives an intuition for whether the neurite is excitatory or not. The scientist evaluated synapse properties close to the cell body in comparison to synapses far away from the cell body (but on the main trunk), and to synapses on far away branches. In this particular case, the study was inconclusive and led him to acquire a bigger data set that will allow him to repeat this analysis with synapses spread out over a longer distance along a dendrite.

Prior to using NeuroLines, the above analysis would have taken our collaborator several hours, if not days to accomplish. He would have started by looking at 2D slices of the segmented EM data set and manually navigate to the locations of synapses, as they were listed in a separate .csv file, to form an initial hypothesis. This step alone is tedious, error-prone, and very frustrating. He would then explain his hypotheses to statisticians working in the same lab, who would program Matlab scripts to try to extract all the necessary information from the data to confirm or refute the theory. In many cases, however, the statistician would find some suspicious outliers or errors in the data. For the scientist this means going back to the 2D slice visualization, manually navigating to the area the statistician had identified and fixing the labeling or meta data, before starting the entire process again. Using NeuroLines, this previous workflow is significantly sped up and simplified. Scientists can directly test and adjust their hypotheses in a single integrated system that allows them to identify errors directly during their analysis. Only once the scientist is sufficiently sure about the hypothesis, the data is given to statisticians for detailed analysis.

10 Discussion

The main qualitative feedback that we received from our collaborators matches our initial intuition: Abstracting the complex and cluttered 3D connectivity patterns into an easy-to-navigate 2D metaphor makes it easier to find patterns in the data, but the link to the original 3D data is still crucial for the scientists. Overall, the scientists felt that their analysis tasks were well supported, and they specifically underlined the added utility of integrating NeuroLines into a larger framework for visual analysis of connectomics data. The best proof of the usefulness of NeuroLines we got by observing our main collaborator. Whenever we showed him a new version of the software he would get an abundance of new hypotheses that he then wanted to verify or refute. He often started with an initial hypothesis but then discovered another interesting pattern that he went on to investigate. This quick turnaround time of forming and evaluating hypotheses was not possible in our collaborators' previous workflow and is exactly what we wanted to enable with NeuroLines. One initially unforeseen, but very useful, feature of NeuroLines is proof-reading segmentation data. Immediately when using the tool, our collaborators would spot and identify segmentation and annotation errors in their data that they had missed before.

The main limitation of NeuroLines with respect to analyzing nanoscale neuronal connectivity is the relatively limited amount of data our collaborators have acquired until now. Although they have acquired terabytes of electron microscopy data, their current segmentation and synapse labeling process is at best semi-automatic and needs a lot of manual input. Therefore, we have evaluated NeuroLines with large synthesized data, and asked our collaborators to navigate within the data set, and to try to find patterns as they would do it in real data. However, this form of scalability testing is still only an approximation of the actual visual analysis of larger data sets in the future.

We believe that NeuroLines successfully addresses the scalability issues present in today's large-scale connectomics data sets, and that our work will also be useful for designing future frameworks for visual connectomics. The combination of abstract information visualization views geared towards analyzing data in an efficient and intuitive way, and traditional volume visualization techniques for exploring the original electron microscope volume and its segmentations, creates a powerful visualization suite that supports different but equally important needs of domain scientists that want to explore large, complex data.

11 Conclusions and Future Work

NeuroLines significantly improves the current workflow of neuroscientists by allowing them to quickly form and test hypotheses in their overall goal of finding out how the brain works. It enables scientists to focus on the connectivity of individual neurites by representing neurites in a subway map-inspired 2D visualization that removes the clutter and complexity of the detailed spatial representations of neurites, while keeping the topology and important spatial cues intact.

We think that the design decisions made while developing NeuroLines will be useful to developers of future visual connectomics or large-scale visualization applications and, furthermore, argue that our neurite visualization metaphor could be applied in more general cases of visualizing interconnected topological trees.

In the future, with the arrival of larger data sets, we would like to incorporate an additional level of abstraction into our visualization, making it possible to analyze the connectivity patterns between entire brain regions. Another interesting path for the future is the comparative analysis of several data sets in a single visualization. For example, our collaborators would like to compare the detailed connectivity patterns of the brain not only between different specimen of the same species, but also between different species, such as a mouse and a monkey. Devising new visualization methodologies that allow scientists to quickly see the differences and similarities in the connectomes of different species would enable interesting new research directions.

Acknowledgments

This work was partially supported by King Abdullah University of Science and Technology, NSF grant OIA-1125087, the NIMH Silvio Conte Center (P50MH094271) and NIH grant 5R01NS076467-04.

References

[1] Eyewire. http://eyewire.org/, 2012. Accessed on 21/07/2014.

[2] J. Anderson, S. Mohammed, B. Grimm, B. Jones, P. Koshevoy, T. Tasdizen, R. Whitaker, and R. Marc. The Viking viewer for connectomics: scalable multi-user annotation and summarization of large volume data sets. *Journal of Microscopy*, 241(1):13–28, 2011.

[3] G. A. Ascoli and J. L. Krichmar. L-neuron: A modeling tool for the efficient generation and parsimonious description of dendritic morphology. *Neurocomputing*, 32-33:1003–1011, 2000.

[4] D. S. Bassett and E. Bullmore. Small-world brain networks. *The Neuroscientist*, 12(6):512–523, 2006.

[5] J. Beyer, A. Al-Awami, N. Kasthuri, J. W. Lichtman, H. Pfister, and M. Hadwiger. ConnectomeExplorer: Query-guided visual analysis of large volumetric neuroscience data. *IEEE TVCG (Proc. IEEE Visualization'13)*, 19(12):2868–2877, 2013.

[6] J. Beyer, M. Hadwiger, A. Al-Awami, W.-K. Jeong, N. Kasthuri, J. W. Lichtman, and H. Pfister. Exploring the connectome: Petascale volume visualization of microscopy data streams. *IEEE Computer Graphics & Applications*, 33(4):50–61, 2013.

[7] M. A. Borkin, K. Z. Gajos, A. Peters, D. Mitsouras, S. Melchionna, F. J. Rybicki, C. L. Feldman, and H. Pfister. Evaluation of artery visualizations for heart disease diagnosis. *IEEE TVCG (Proc. InfoVis'11)*, 17(12):2479–2488, 2011.

[8] S. Bremm, T. von Landesberger, M. Hess, T. Schreck, P. Weil, and K. Hamacherk. Interactive visual comparison of multiple trees. In *IEEE VAST*, pages 31–40. IEEE, 2011.

[9] K. L. Briggman and W. Denk. Towards neural circuit reconstruction with volume electron microscopy techniques. *Current Opinion in Neurobiology*, 16(5):562–570, 2006.

[10] S. Bruckner, V. Šoltészová, M. E. Gröller, J. Hladůvka, K. Bühler, J. Yu, and B. Dickson. BrainGazer - visual queries for neurobiology research. *IEEE TVCG (Proc. IEEE Visualization'09)*, 15(6):1497–1504, 2009.

[11] E. Bullmore and O. Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, 2009.

[12] A. Cardona, S. Saalfeld, J. Schindelin, I. Arganda-Carreras, S. Preibisch, M. Longair, P. Tomancak, V. Hartenstein, and R. J. Douglas. TrakEM2 software for neural circuit reconstruction. *PLoS ONE*, 7(6), 06 2012.

[13] R. C. Coelho, N. Calonego Jr., and L. A. Consularo. Visualization and simulation of 3D artificial neural structures generated by L-system. *Virtual Real.*, 14(2):119–129, 2010.

[14] W. M. Fitch and E. Margoliash. Construction of phylogenetic trees. *Science*, 155:279–284, 1967.

[15] S. Gerhard, A. Daducci, A. Lemkaddem, R. Meuli, J. Thiran, and P. Hagmann. The connectome viewer toolkit: An open source framework to manage, analyze, and visualize connectomes. *Frontiers in Neuroinformatics*, 5, 2011.

[16] M. Graham and J. Kennedy. A survey of multiple tree visualisation. *Information Visualization*, 9(4):235–252, 2010.

[17] M. Hadwiger, J. Beyer, W.-K. Jeong, and H. Pfister. Interactive volume exploration of petascale microscopy data streams using a visualization-driven virtual memory approach. *IEEE TVCG (Proc. IEEE SciVis'12)*, 18(12):2285–2294, 2012.

[18] D. Haehn, S. Knowles-Barley, M. Roberts, J. Beyer, N. Kasthuri, J. Lichtman, and H. Pfister. Design and evaluation of interactive proofreading tools for connectomics. *IEEE TVCG (Proceedings of SciVis'14)*, to appear, 2014.

[19] M. Helmstädter, K. L. Briggman, and W. Denk. High-accuracy neurite reconstruction for high-throughput neuroanatomy. *Nature Neuroscience*, 14(8):1081–1088, 2011.

[20] A. Irimia, M. C. Chambers, C. M. Torgerson, and J. D. V. Horn. Circular representation of human cortical networks for subject and population-level connectomic visualization. *NeuroImage*, 60(2):1340 – 1351, 2012.

[21] W.-K. Jeong, J. Beyer, M. Hadwiger, R. Blue, C. Law, A. Vázquez-Reina, C. Reid, J. W. Lichtman, and H. Pfister. Ssecrett and neurotrace: Interactive visualization and analysis tools for large-scale neuroscience datasets. *IEEE Computer Graphics & Applications*, 30(3):58–70, 2010.

[22] R. Jianu, C. Demiralp, and D. Laidlaw. Exploring brain connectivity with two-dimensional neural maps. *IEEE TVCG*, 18(6):978–987, 2012.

[23] A. Kanitsar, D. Fleischmann, R. Wegenkittl, P. Felkel, and M. Gröller. CPR - curved planar reformation. In *IEEE TVCG*, pages 37–44, 2002.

[24] V. Kaynig, A. Vázquez-Reina, S. Knowles-Barley, M. Roberts, T. Jones, N. Kasthuri, E. Miller, J. W. Lichtman, and H. Pfister. Large-scale automatic reconstruction of neuronal processes from electron microscopy images. In *arXiv: 1303.7186 [q-bio.NC]*, 2013.

[25] S. Knowles-Barley, M. Roberts, N. Kasthuri, D. Lee, H. Pfister, and J. W. Lichtman. Mojo 2.0: Connectome annotation tool. *Frontiers in Neuroinformatics*, (60), 2013.

[26] T.-C. Lee, R. L. Kashyap, and C.-N. Chu. Building skeleton models via 3-D medial surface/axis thinning algorithms. *CVGIP: Graph. Models Image Process.*, 56(6):462–478, 1994.

[27] A. Lex, C. Partl, D. Kalkofen, M. Streit, S. Gratzl, A. M. Wassermann, D. Schmalstieg, and H. Pfister. Entourage: Visualizing relationships between biological pathways using contextual subsets. *IEEE TVCG (Proc. IEEE InfoVis'13)*, 19(12):2536–45, 2013.

[28] A. Lex, M. Streit, E. Kruijff, and D. Schmalstieg. Caleydo: Design and evaluation of a visual analysis framework for gene expression data in its biological context. In *Proc. of IEEE Pacific Visualization Symposium (PacificVis'10)*, pages 57–64. IEEE, 2010.

[29] K. Li, L. Guo, C. Faraco, D. Zhu, H. Chen, Y. Yuan, J. Lv, F. Deng, X. Jiang, T. Zhang, X. Hu, D. Zhang, L. S. Miller, and T. Liu. Visual analytics of brain networks. *NeuroImage*, 61(1):82 – 97, 2012.

[30] J. W. Lichtman and W. Denk. The big and the small: Challenges of imaging the brain's circuits. *Science*, 334(6056):618–623, 2011.

[31] C.-Y. Lin, K.-L. Tsai, S.-C. Wang, C.-H. Hsieh, H.-M. Chang, and A.-S. Chiang. The neuron navigator: Exploring the information pathway through the neural maze. In *IEEE Pacific Visualization Symposium (PacificVis'11)*, pages 35–42, 2011.

[32] D. S. Margulies, J. Böttger, A. Watanabe, and K. J. Gorgolewski. Visualizing the human connectome. *NeuroImage*, 80(0):445 – 461, 2013. Mapping the Connectome.

[33] R. Miller, V. Mozhayskiy, I. Tagkopoulos, and K.-L. Ma. Evevis: A multi-scale visualization system for dense evolutionary data. In *IEEE Symp. on Biological Data Vis. (BioVis'11)*, pages 143–150, Oct 2011.

[34] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou. TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'03)*, 22(3):453–462, 2003.

[35] K. Nesbitt. Getting to more abstract places using the metro map metaphor. In *8th International Conference on Information Visualisation (IV'04)*, pages 488–493, 2004.

[36] H. Pfister, V. Kaynig, C. P. Botha, S. Bruckner, V. J. Dercksen, H.-C. Hege, and J. B. T. M. Roerdink. Visualization in connectomics. In *arXiv:1206.1428v2 [cs.GR]*, 2012.

[37] M. Roberts, W.-K. Jeong, A. Vázquez-Reina, M. Unger, H. Bischof, J. W. Lichtman, and H. Pfister. Neural process reconstruction from sparse user scribbles. In *Proc. of MICCAI 2011*, pages 621–628. Springer, 2011.

[38] S. Saalfeld, A. Cardona, V. Hartenstein, and P. Tomančák. CATMAID: collaborative annotation toolkit for massive amounts of image data. *Bioinformatics*, 25(15):1984–1986, 2009.

[39] H. Schulz. Treevis.net: A tree visualization reference. *IEEE Computer Graphics & Applications*, 31(6):11–15, 2011.

[40] S. Seung. *Connectome: How the Brain's Wiring Makes Us Who We Are*. Houghton Mifflin Harcourt, 2012.

[41] A. Sherbondy, D. Akers, R. Mackenzie, R. Dougherty, and B. Wandell. Exploring connectivity of the brain's white matter with dynamic queries. *IEEE TVCG*, 11(4):419–430, 2005.

[42] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. of the IEEE Symposium on Visual Languages*, VL '96, pages 336–343. IEEE, 1996.

[43] J. Sorger, K. Bühler, F. Schulze, T. Liu, and B. Dickson. neuroMap - interactive graph-visualization of the fruit fly's neural circuit. In *IEEE Symp. on Biological Data Vis. (BioVis'13)*, pages 73–80, Oct 2013.

[44] J. Stott, P. Rodgers, R. Burkhard, M. Meier, and M. Smis. Automatic layout of project plans using a metro map metaphor. In *Proc. Information Visualisation'05*, pages 203–206. IEEE, 2005.

[45] A. Telea. Combining extended table lens and treemap techniques for visualizing tabular data. In *Proc. Eurographics/IEEE VGTC Conference on Visualization (EuroVis'06)*, pages 51–58. Eurographics Association, 2006.

[46] F. van Ham and A. Perer. "Search, show context, expand on demand": supporting large graph exploration with degree-of-interest. *IEEE TVCG (Proc. InfoVis'09)*, 15(6):953–60, 2009.

[47] D. Watts and S. Strogatz. Collective dynamics of "small-world" networks. *Nature*, (393):440–442, 1998.