

# NeuroBlocks – Visual Tracking of Segmentation and Proofreading for Large Connectomics Projects

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

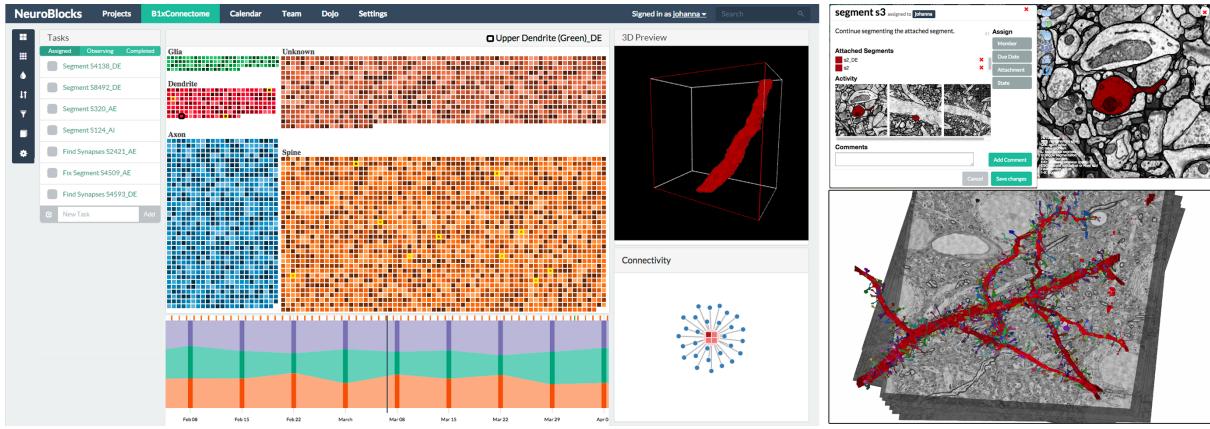


Fig. 1. *Visualizing large-scale segmentation projects in connectomics.* Left: The main view of NeuroBlocks, comprising a scalable “pixel-based” visualization of the current segmentation state on top, and a timeline for exploring the project’s evolution at the bottom. Top right: Detailed view of a segmentation task assigned to a user, showing recent modifications and thumbnails of the corresponding tool states. Bottom right: Linked petascale volume rendering of the underlying microscopy data and the selected segmented object.

**Abstract**—In the field of connectomics, neuroscientists acquire electron microscopy volumes at nanometer resolution in order to reconstruct a detailed wiring diagram of the neurons in the brain. The resulting image volumes, which often are hundreds of terabytes in size, need to be segmented to identify cell boundaries, synapses, and important cell organelles. However, the segmentation process of a single volume is very complex, time-intensive, and usually performed using a diverse set of tools and many users. To tackle the associated challenges, this paper presents NeuroBlocks, which is a novel visualization system for tracking the state, progress, and evolution of very large volumetric segmentation data in neuroscience. NeuroBlocks is a multi-user web-based application that seamlessly integrates the diverse set of tools that neuroscientists currently use for manual and semi-automatic segmentation, proofreading, visualization, and analysis. NeuroBlocks is the first system that integrates this heterogeneous tool set, providing crucial support for the management, provenance, accountability, and auditing of large-scale segmentations. We describe the design of NeuroBlocks, starting with an analysis of the domain-specific tasks, their inherent challenges, and our subsequent task abstraction and visual representation. We demonstrate the utility of our design based on two case studies that focus on different user roles and their respective requirements for performing and tracking the progress of segmentation and proofreading in a large real-world connectomics project.

**Index Terms**—Neuroscience, Segmentation, Proofreading, Data and Provenance Tracking

## 1 INTRODUCTION

In recent years, connectomics has become a very promising subfield of neuroscience, with the aim to reconstruct the complete wiring diagram of the mammalian brain at nanometer resolution. To achieve this goal, high-resolution electron microscopy (EM) volumes of brain

tissue are acquired at a rate above 10 to 40 megapixels per second [8], and subsequently have to be segmented, labeled, and analyzed.

Because of this, image and volume segmentation have become a major bottleneck in connectomics research. Successfully tackling the associated challenges requires a variety of tools as well as scalable visualization support. One area where visual representations and interactive visualization tools can be especially helpful is the integration of a wide variety of heterogeneous components, tasks, tools, and data types into a coherent, efficient workflow that is driven by interactive visualization. Currently, in neuroscience no existing tool targets this goal, which is therefore the aim of the work described in this paper.

Many tools used in current practice in neuroscience are still mostly based on performing manual segmentation, which is a very labor- and time-intensive process. Recently, a lot of research in computer vision and image processing has focused on developing (semi-) automatic segmentation methods for extracting cell boundaries and specific cell organelles from EM data. However, while fully automatic approaches in principle do not require any user interaction, they still exhibit higher error rates than manual segmentation. For this reason, a subsequent semi-automatic *proofreading* step is usually still necessary in practice, which again requires user interaction.

- 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, Daniel Haehn and Hanspeter Pfister are with the School of Engineering and Applied Sciences at Harvard University.  
*E-mail:* {jbeyer, haehn, pfister}@seas.harvard.edu.
- Narayanan Kasthuri is with the School of Medicine at Boston University.  
*E-mail:* bkasthur@bu.edu.
- Jeff W. Lichtman is with the Center for Brain Science at Harvard University.  
*E-mail:* jeff@mcb.harvard.edu.

*Manuscript received 31 Mar. 2015; accepted 1 Aug. 2015; date of publication xx Aug. 2015; date of current version 25 Oct. 2015.*

*For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.*

To keep up with the fast increasing amount of raw data, neuroscientists now routinely manage a whole team comprising *segmenters* and *proofreaders*. They divide the segmentation tasks between these users, whose experience ranges from senior researchers to novice interns. This in turn makes managing and keeping track of the current segmentation data a very complex task. Users not only have different levels of experience, but might also be working remotely and at different locations. Furthermore, mentally keeping track of recent segmentation changes, double-checking and approving the changes of inexperienced users, as well as managing their workload and specific tasks has become a major issue that neuroscientists have to deal with.

Managing such a large-scale segmentation project is further complicated by the fact that a typical neuroscience lab uses a plethora of different tools for registration, segmentation, labeling, visualization, and analysis of the data. However, most of the time interfaces between different tools are non-existent. For example, navigating to the same spatial position in two different applications might require manually typing in the position that was extracted from one tool into the user interface of another tool. This also complicates keeping track of the current segmentation state, as switching between different tools is usually slow and cumbersome. Moreover, not all tools support the same kind of meta data, so interoperability becomes very difficult.

Therefore, it is an important goal to design a tool that supports neuroscientists in tracking the progress of very large volumetric segmentation data by offering visual abstractions and scalable views that makes tracking such a large amount of data feasible. Additionally, this tool needs to inherently support multiple users and different user groups, and should seamlessly integrate the diverse set of tools that are used in practice. At the same time, the visualization of the original microscopy data and the segmentation data in 2D and 3D is crucial, and needs to be tightly integrated into the overall workflow.

**NeuroBlocks.** We present a novel visualization system called NeuroBlocks, for tracking the state, progress, and evolution of large segmented EM volumes. NeuroBlocks is a multi-user web-based application that supports provenance, accountability, and auditing of segmentation results. It tightly integrates 2D and 3D interactive visualizations of image and volume data, displays connectivity, and offers abstract data visualization views of the current and former segmentation states. It allows users to see, track, and manage segmentation data that are changing over time, together with the tools that were used in each step. It enables users to go back to any point of interest in time, and to the corresponding state of the tool used at that point.

**Contributions.** Our first contribution is the definition and discussion of a set of requirements and design guidelines for visually managing large-scale segmentation projects in the domain of neuroscience, which can be seen as a specific case of task progress management.

Our second contribution is the design of scalable visual representations for large segmentation data. Our visualizations are fully interactive, and allow users to track the current state, evolution, and progress of the on-going segmentation and proofreading tasks and projects.

Our third contribution is a direct integration of multi-user support—including different user groups with varying permissions, experience, and assigned tasks—into the visualization. NeuroBlocks allows auditing and crediting of users, and provides accountability of who did what and when. Furthermore, we enable users to seamlessly switch between different (segmentation) tools and their respective application states.

Finally, our fourth contribution is a demonstration of the utility of our design based on two case studies. The first case study is performed by a domain expert, who manages and tracks a large segmentation project. The second case study is performed by a relatively inexperienced proofreader, who uses NeuroBlocks to complete the assigned tasks and to quickly switch between different segmentation tools.

## 2 RELATED WORK

**Connectomics.** The main goal of connectomics is reverse-engineering the brain and its neuronal connectivity through the use of high-resolution and high-throughput imaging techniques. Introductions to connectomics are given by Lichtman and Denk [27] and Seung [33].

**Segmentation and proofreading for connectomics.** Before the neuronal connectivity can be extracted from EM data, correct annotation and segmentation of the acquired data volumes is necessary, which is currently one of the main bottlenecks in the connectomics pipeline. Despite its obvious limitations and lack of scalability, manual annotation [3, 5, 12, 30] is still widely used in practice. More recently, semi-automatic segmentation approaches [21, 24, 37] as well as fully automatic methods for segmenting neuronal structures [20, 22] have received a lot of attention. While these automatic approaches are more scalable than manual segmentation and require only little or no user input, they still often need an additional proofreading step, where erroneous segmentations are fixed in a semi-automatic fashion [17, 25].

To handle huge amounts of EM and segmentation volumes, as well as additional annotations and labels, Burns et al. [11] recently presented the open connectome project, a scalable data infrastructure for annotation and analysis of high-throughput brain imaging data.

**Visualization in connectomics.** Most visualization tools for connectomics focus either on displaying the abstract connectivity between cells, or on displaying the original image/volume and segmentation data. The Viking Viewer [3] represents neurons as nodes in an abstract connectivity graph. NeuroMap [38] makes use of circuit wiring diagrams to represent all possible connections of neurons. NeuroLines [1] uses a relative distance-preserving subway map metaphor to visualize and explore neuronal connectivity at the level of individual synapses.

For exploring petavoxel EM data, Hadwiger et al. [16] have presented a volume rendering pipeline that is entirely driven by the visibility of volume bricks in the final image, which was later extended to segmented volumes [7]. BrainGazer [9] and ConnectomeExplorer [6] focus on interactive and visual queries to explore brain connectivity. We have integrated the latter framework into NeuroBlocks as one of several available options for 3D visualization.

However, all of these visualization applications assume a fixed segmentation state. They do not show the progress and evolution of large-scale segmentation volumes and projects.

**Provenance and progress tracking.** Capturing provenance data in computational tasks (e.g., visualization systems) is crucial to make data exploration, processing, and analysis reproducible [13]. The VisTrails system [4] is an open-source framework that tracks and stores provenance data for visualization pipelines. It not only captures data provenance, but also provenance of the user’s exploration process. VisTrails and its provenance management have been incorporated into many different applications, ranging from creating visualizations by analogy and query-by-example [32], to provenance-aware workflows for oceanography [19], and reproducible scientific experiments [14].

Other provenance systems include Halaschek et al. [18], who track provenance for annotation and management of images on the semantic web, while Simmhan et al. [36] track the derivation history of data for provenance in data-driven workflows. More recently, Al-Naser et al. [2] employ provenance in the context of seismic visualization to manage multi-user and multi-version seismic interpretations. However, none of these systems have tackled provenance for large segmentation data in complex, distributed workflows.

**Pixel views and comparative visualization.** The goal of pixel views or dense information views is to represent as many data objects as possible within the available screen space. Keim [23] discusses design decisions for pixel-oriented views for visualizing large multi-dimensional data sets, including different pixel arrangements, such as Morton curves or query-based ordering. A common two-dimensional space-filling approach for representing tree structures are Treemaps [34] and their extension to squarified treemaps [10]. Heatmaps are another type of dense information visualization, and are very popular in bioinformatics and genomics [26].

OnSet [31] is a space-filling visualization of binary set data that supports comparison between and combination of sets, and scales to several hundred elements per set. Gleicher et al. [15] have presented a general survey on comparative visualization. They introduce a general taxonomy of visual design for comparisons based on juxtaposition, superposition, and explicit encoding. More recently, Wang et al. [40] have proposed a treemap approach for displaying similarity-based im-

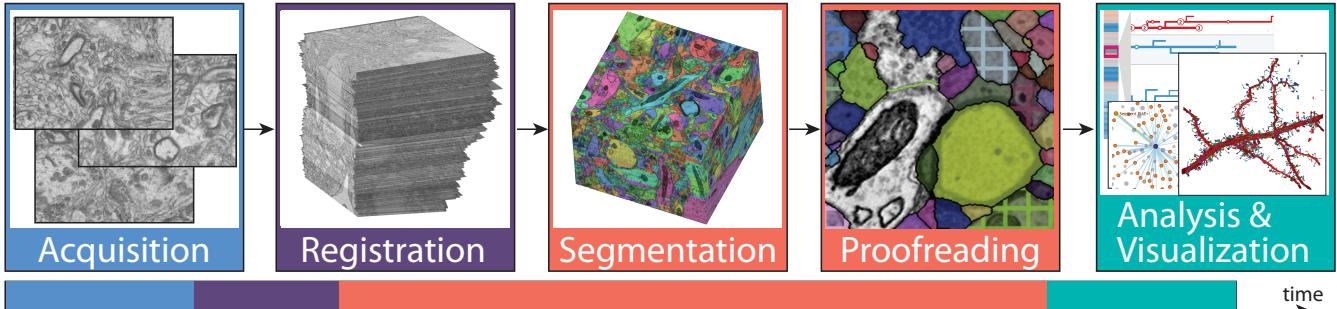


Fig. 2. **Connectomics workflow.** Top: High-resolution electron microscope images are acquired and registered into a 3D volume before neuronal structures are segmented and labeled. Automatic segmentation approaches require a subsequent proofreading step before the data can be visualized and analyzed. Bottom: Visualization of the typical relative durations of these tasks in a medium- to large-scale neuroscience project.

age collections on large tiled displays. Lindemann et al. [28] have presented a volume visualization system for comparative visualization of segmentation masks from different modalities or different timesteps. However, they focus on medical data comprising only a small number of segmented objects, whereas in our target area of connectomics we have to deal with thousands of objects or more in a single data set.

**Multi-scale and multiple linked views.** Efficient means for exploring large numbers of data elements are hierarchical navigation metaphors and focus-and-context techniques [35]. In the context of connectomics, NeuroLines [1] uses a multi-level navigation metaphor, with the lowest level inspired by heat maps and extended table lenses [39] for multi-column sorting. Lex et al. [26] use a multi-tier focus-and-context approach for large heat maps in genomics.

An alternative approach to linking views—or even entire applications—together was presented in ManyVis [29], which is a framework that can combine different independent visualization tools into a single integrated application by intercepting and processing low-level user interactions. ManyVis was also used to implement rapid annotation of large microscopy images, with the focus of integrating an out-of-core visualizer into Adobe Photoshop. However, ManyVis does not incorporate provenance or tracking of segmentation changes.

### 3 BASIC DESIGN CONSIDERATIONS

We first discuss the major design considerations for NeuroBlocks, starting with the necessary neuroscience background, then describing the general goals, domain-specific tasks, and challenges we have identified. Finally, we motivate our design decisions and considerations, including lessons learned from previous alternative design prototypes.

#### 3.1 Neuroscience Background and Workflow

The connectomics workflow of our collaborators starts with high-throughput electron microscopy image acquisition of tiny samples of mouse brain tissue. This results in image tiles with a resolution of 3-5 nm per pixel, and a slice thickness of 25-30 nm. Before further data processing, the image tiles are stitched and registered to form a single 3D volume with a slice resolution of 20,000 to 100,000 pixels, and up to tens of thousands of slices. The next step in the pipeline consists of segmenting and annotating the neural structures in the data, before visualization and final analysis can be performed (see Figure 2).

In connectomics, the main goal is to construct a wiring diagram of how individual *neurons* (i.e., nerve cells) are connected in the brain. Neurons transmit electrical signals to neighboring neurons over *synapses*, the connections between neurons. The size and form of neurons can vary widely, but typically they consist of several *dendrites*, tree-like branching structures that receive input from other neurons, and an *axon*, a long and narrow tubular structure that transmits signals away from the cell body towards other neurons. Additionally, neurons have several sub-structures or cell organelles, and even synapses have additional discriminating features, such as vesicles, boutons, or spines.

The high structural variability of neurons combined with the low signal-to-noise ratio of EM images make segmentation of this data type challenging. Therefore, and because of the large data sizes, segmentation is often the bottleneck of the entire connectome workflow.

Our collaborators use a wide variety of tools for registration, segmentation, visualization, and analysis. Almost every researcher has their own favorite tools, often very specific to the task they want to perform. Currently, our collaborating neuroscientists routinely use VAST [5] for manual segmentation, and the RhoANA pipeline [22] for automatic segmentation. The RhoANA pipeline performs state-of-the-art automatic reconstruction of neuronal processes in terabyte EM data based on a random forest classifier coupled with an anisotropic smoothing prior in a conditional random field framework and 3D segment fusion. Our collaborators typically generate a slightly over-segmented dataset to simplify and speed up the semi-automatic proofreading step, which is performed in Mojo [25] and Dojo [17]. During this proofreading step, additional data and metadata can be generated, such as synapse labels (stored as  $x, y, z$  positions), textual labels, or segment names. For visualization and analysis, our collaborators use Matlab, ConnectomeExplorer [6], and NeuroLines [1].

#### 3.2 Task Analysis

The design of NeuroBlocks was developed in close collaboration with neuroscientists working in the field of connectomics. Over the course of several months we have conducted detailed informal as well as semi-structured interviews with four domain scientists focusing on EM-based connectomics research: two grad students, one postdoctoral researcher, and one faculty. Our collaborators typically work on large-scale segmentation projects, with data sets ranging up to hundreds of terabytes in size. Although oftentimes these data do not have to be segmented in their entirety, the required effort is still so large that many undergrads and interns are employed as segmenters or proofreaders, which places a significant management burden on the scientists.

A common finding in our interviews was that even though the ultimate goal of the neuroscientists is the analysis of their data, most of their time was spent on segmenting, labeling, and proofreading data (see Figure 2, bottom). Interestingly, even though three of them initially worked at the same lab, they all used different combinations of segmentation and visualization tools, and there was no “standard workflow.” For example, switching from segmentation tools to visualization or analysis tools was mostly a hand-tuned, improvised solution and sometimes required several hours to export the data from one tool into the next. Keeping track of the current and previous states of the segmentation was not supported by any of their tools, even though our collaborators considered this to be one of the most cumbersome tasks.

##### 3.2.1 Domain Goals

The ultimate goal of our collaborators is doing research based on analysis of their segmented data. However, in order to achieve this, their immediate goal for NeuroBlocks is having a framework for managing, verifying, and tracking the state of individual segmentation projects.

This includes (a) seeing the current segmentation, (b) tracking its evolution, (c) managing multiple users with different levels of experience and having an overview of their current progress, (d) creating subtasks and tracking their progress, and (e) being able to look at the segmentation in 2D and 3D with different tools, and moreover being able to seamlessly switch between them.

For example, a scientist might assign four segmenters to four different proofreading tasks. The segmenters do not have a lot of experience, so the scientist has to individually approve all of their changes (e.g., by looking at a *diff* image in 2D or 3D), before they are merged into the main segmentation volume. Once these tasks are completed, the scientist can use NeuroBlocks to quickly extract a meaningful subset of the segmented data for a preliminary analysis of the data, or to generate a progress report of the current state of the segmentation.

### 3.2.2 Domain-Specific Tasks

We have identified the following domain tasks for managing large-scale segmentation projects in neuroscience:

**T1 - Overview and detail visualization of the current segmentation state.** Dense segmentations of large EM volumes can contain hundreds of thousands of objects. Neuroscientists need to be able to get an overview of the entire data set, while also being able to look at individual segments in detail. A combination of 2D and 3D visualizations of the original volume data for highly detailed views in combination with visual abstractions for higher-level contextual information is required to fulfill this task.

**T2 - Track changes to the segmentation volume.** Especially in large-scale segmentation projects, with multiple people collaborating on the same data set, it is necessary to not only look at the current state of a segmentation, but also at previous states and their evolution.

**T3 - Manage fine-grained segmentation tasks.** Being able to discuss task-specific details, and to collect and visualize provenance of the used tools for a specific task, are crucial for accountability of segmentations, and for coordinating the tasks of multiple users.

**T4 - Audit users and segmentations.** The importance of provenance and accountability for creating scientific results based on a user's segmentation requires the possibility to audit and verify individual segmentations created by a specific user.

**T5 - Switch seamlessly between different segmentation and visualization tools.** One of the main unnecessary bottlenecks of our collaborating scientists is when they have to switch from one tool to another one. Therefore, it is crucial to develop a framework that seamlessly integrates the different tools that are currently used, while also planning for easy future extensibility.

### 3.2.3 Challenges

The goals and domain-specific tasks described above directly correspond to some of the challenges of designing a framework for managing large-scale segmentation projects. One of the biggest fundamental challenges is achieving scalability: with respect to overall data size, the number of segmentations, the number of segmentation changes, the number of current tasks, and the number of users.

Typical projects comprise many, often distributed users of different experience levels in segmentation and proofreading. The current strategy of our collaborators is to hire high-school interns for proofreading tasks. Naturally, not all interns will create segmentations of equally high quality. Therefore, multiple hierarchical levels of checking and approving changes to the segmentation need to be employed.

Finally, the diverse eco-system of the current segmentation and visualization tools available for connectomics data makes it difficult to find a common standard that is supported by the most common tools.

## 3.3 Design Iterations

Our collaborators were initially interested in having something similar to a virtual lab notebook, which would allow them to track their progress and jot down notes in a semi-structured fashion. Our first design was heavily based on this idea, and consisted of a virtual notebook that also allowed users to dynamically integrate external visualization and analysis applications into their notebooks (see Figure 3 (a)).

This was well received and we subsequently focused more on the aspect of provenance, how to track the current analysis, and how to quickly explore similar analysis or visualization options. An initial design of this stage is shown in Figure 3 (b). It includes a provenance tree of different analysis states.

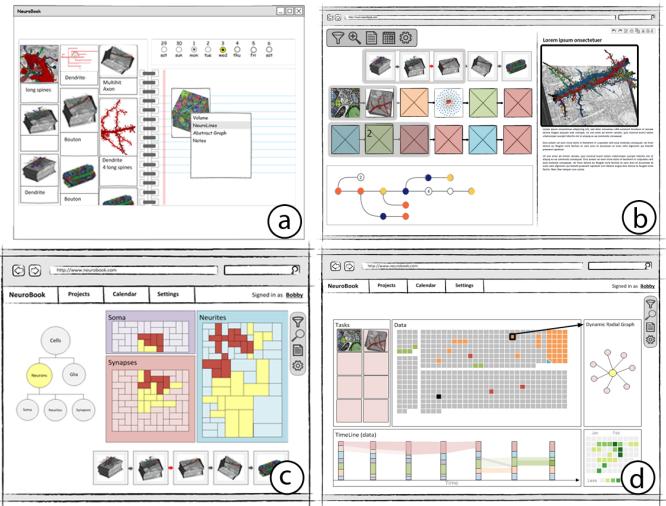


Fig. 3. *Design iterations of NeuroBlocks.* (a) virtual lab notebook; (b) provenance-driven analysis; (c) data-centric view; (d) final design.

However, over the course of several meetings with our domain scientists it became obvious that their current main bottleneck was handling the segmentation part of their project, not the visualization by itself and the actual analysis. This means that for the major part of their projects they are mostly interested in the current state of their data and its segmentation. Only in the later stages of a project are they interested in the state of their analysis and visualization.

We therefore adjusted our objective from an analysis-centric view on their project to a data-centric view on their project, see Figure 3 (c). This allowed the scientists to focus on the current segmentation and leverage a data abstraction based on specific neuroscience objects (e.g., axons, dendrites). This shift to a data-centric view drastically expanded the way our collaborators thought about the project, and was the single most important design decision during the course of the project. However, they found the initial treemap design with variable sized elements confusing and preferred a fixed size for all elements.

A sketch of the final design of NeuroBlocks is shown in Figure 3 (d), with a main data view, a timeline, a task list and a detail-on-demand view for showing details of a selected element in the data view. By using visualization not just for the analysis and exploration of the final data, but already for tracking their segmentation progress, our collaborators now have a much better grasp of their data, of the parts that are still missing and need to be identified and segmented, and they are more aware of potential problems in their segmentations.

## 4 VISUAL DESIGN

The final visual design that we have implemented comprises several linked views. Figure 1 shows the main elements of a segmentation project in NeuroBlocks: a segmentation state view, a timeline, and different views for details-on-demand. Additionally, NeuroBlocks integrates 2D and 3D views of the original EM volume and segmentation data (Figure 1, right), as well as ‘diff’ views of segmentation changes.

### 4.1 Visualizing the Current Segmentation State

The objective for visualizing the current segmentation state is to provide a dense overview of the available segmentation data. To guarantee scalability to tens of thousands of segmented objects, the provided visualization abstracts and simplifies the underlying volumetric data significantly, but details can always be shown on demand.

#### 4.1.1 Segmentation State Pixel View

The main view of NeuroBlocks is the data-centric *segmentation state pixel view*. To achieve scalability with respect to the number of segments in the segmentation project, we chose a pixel-based representation of the data, where each visual element represents one segment. We call this visual element a *pixel*, even though it is usually larger than

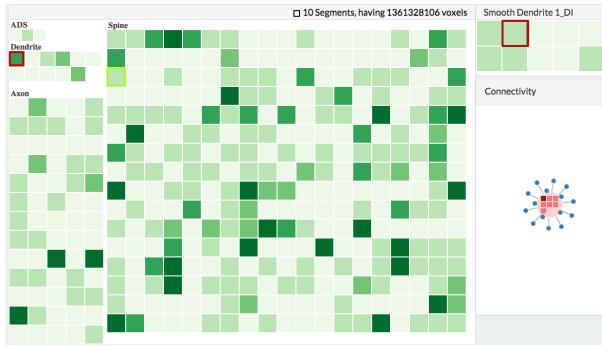


Fig. 4. *Multi-scale pixel view of the current segmentation state.* “Downsampling” the full-resolution pixel view ensures overview visibility of all data. Here, as an example, elements are sorted by size, and color-coded according to the number of recent modifications. All individual segments comprising the selected super-pixel are shown in the top right. The bottom right shows a connectivity graph of the selected object.

a screen pixel. This visual abstraction enables us to create a compact overview visualization that is dense in information and can show thousands of individual segments at the same time in the available screen space. Details of individual pixels can be shown on demand and explored in the multiple linked views (see Sec. 4.3).

In the default state, the segmentation state pixel view displays the *current state* of a segmentation project. Individual segments are shown as pixels and are structured in a hierarchy, according to a neuroscience taxonomy that our collaborators follow when labeling the data. For example, a segment can be categorized as  $segment \in spine \in dendrite \in neuron$ . Several segments can belong to the same *segmentation object*, and alternatively we can map entire segmentation objects to individual pixels, instead of mapping individual segments.

#### 4.1.2 Sorting, Filtering and Layouting

Pixels in the pixel view are sorted, filtered, and highlighted based on user input. Default settings can be specified for each NeuroBlocks project separately. For example, in a project that focuses on proofreading of automatically segmented data, the project manager would get a view where pixels are sorted based on proofreading completion status with a color-coding that shows the number of modifications in a segment. This enables directly getting an overview of the current progress. Other options for sorting and filtering include showing only segments that were modified by a certain user, sorting by segment size, or highlighting all segments that were modified since the user last logged in. Pixels are always sorted from left to right and top to bottom, to correspond to the general reading direction. The individual pixel size is either set to a constant, or adjusts automatically based on the number of available segments. To ensure that the pixel size never falls below a certain threshold, a multi-scale representation is displayed if there are too many pixels for the current display (see Sec. 4.1.3).

The pixel view supports three main layouts: Pixels can either be displayed in a nested hierarchy, as treemap [34], or in a flat view. The nested hierarchy is shown in Figure 1, and clusters segments based on their assigned neuroscience category. This category is typically assigned either during segmentation or proofreading. The treemap view also shows the data clustered based on segment category, but avoids any white space. The flat view avoids any clustering, and displays all segments as a single block. This is particularly useful for fast sorting and filtering over all segments, and for uncategorized data.

#### 4.1.3 Multi-Scale Visualization

Automatic segmentation methods typically over-segment data because merging segments in the subsequent proofreading step is easier and faster than splitting segments [17]. This, however, often results in a large number of individual segments, exceeding the available screen space in our pixel view. Therefore, we developed a multi-scale pixel-based view, guaranteeing visibility of the entire data set on screen.

The multi-scale view is computed based on the sorting order of the segments, and recursively combines a pre-defined number of pixels

into a single “super-pixel.” Conceptually, this is similar to downsampling. However, we can specify the number of elements that are combined into a super-pixel as well as the type of aggregation operation (e.g., average, min, max). Figure 4 shows a multi-scale pixel view with the coloring highlighting the recent changes in the segmentation.

## 4.2 Visualizing the Segmentation History and Evolution

In addition to showing a single timestep or snapshot of the segmentation, NeuroBlocks also supports exploring the evolution or provenance of the segmentation. In contrast to most provenance visualization systems, we do not focus on a graph-based visualization of provenance data, but chose a timeline-based approach for navigating between different segmentation states. This has the advantage that it is easy to navigate at multiple time scales, and that it is intuitive for novice users.

Exploring the evolution of the segmentation is supported in three different ways: First, by moving a slider through a timeline (see Sec. 4.2.1), users can navigate through the segmentation history. In this mode, the pixel view is synced to the time slider and shows the segmentation state at the position of the time slider. While moving the slider, the pixel view transitions between the different time steps in a smooth animation. This allows users to intuitively follow the evolution of the segmentation and see newly added segments or track how segments change size over time. Second, the user can choose two timestamps, and the pixel view will then show a direct comparison between these two states. This highlights the changes that have occurred between two distinct time steps. Segments can be color-coded and sorted based on different criteria, such as change in size, or showing the segments whose categorization changed in the specified time interval. Finally, NeuroBlocks offers a multi-scale time navigation. For a specified time span or time interval, the pixel view can display the frequency of change in segments. This time interval can be resized and moved interactively, similarly to the time slider.

### 4.2.1 Timeline View

In addition to navigating between different time steps and temporal intervals via time sliders and brushes, the *timeline view* can also directly visualize attributes over time. While the former can be used to explore the evolution of a segmentation project, and to explore the project in different time scales and intervals by using linking and brushing, the latter offers a direct visualization of user specified attributes over time, which are displayed as stacked area charts. The default attributes that are shown vary depending on the type of project and the current user. For example, it can show recent activity of all users, the amount of change in segments over time, or task status. The timeline in Figure 1 shows the activity of different users, and how many segments they have modified recently. The timeline is implemented as a dynamic scale and allows users to switch between highly detailed views of individual days or hours, and coarse views of entire weeks or months.

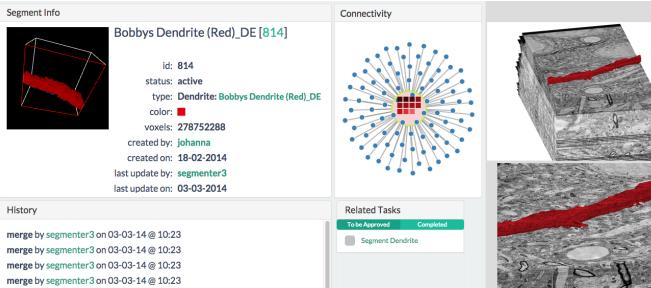
## 4.3 Visualizing Details on Demand

Once the user has examined the high-level structure of a segmentation project, NeuroBlocks offers different views for exploring the details of individual objects, tasks, users, and segments. Details are typically shown on demand, whenever a segment is selected in the pixel view.

### 4.3.1 Segment and Object Detail Views

Individual segments and objects (i.e., semantic structures that consist of more than one segment) can be examined in more detail in the *segment view* and the *object view*, respectively. Figure 5 displays a segment detail view, including an interactive 3D preview of the selected segment (top left), the segmentation history (bottom left), and a connectivity view (top, second from right). Figure 5 (right) shows the selected segment in a ConnectomeExplorer 3D volume view.

The object and segment detail views allow users to explore previous edits to the segmentation, and allow senior users to approve or reject a proposed segmentation change. Additionally, the previous segmentation states of the selected segment can be displayed as thumbnails, and allow users to directly go back to that state in the corresponding segmentation tool by clicking on the thumbnail.



**Fig. 5. Segmentation object details and volume rendering integration.** Details about the segmented object, all its segments, revision history, and collaborators are shown in this view. Recent changes to the segmentation can be examined in detail and at full resolution in the integrated volume renderer, and then be approved or discarded by the user. The view at the bottom right shows a “diff” of the most recent merge.

#### 4.3.2 Connectivity View

Neuronal objects such as axons or dendrites are typically very highly interconnected. Therefore, NeuroBlocks offers an abstract object visualization that shows which segments are part of an object, and a node-link diagram showing the directly connected neighbors (Figure 5).

The center of the connectivity view shows the selected object as a circular node, and all the segments it contains as pixels within that object. Whenever a pixel is selected in the main pixel view, the connectivity view gets updated and the selected segment is highlighted in both views respectively. Connectivity of objects is based on the labeled synapses an object makes to neighboring cells. These connected objects are displayed in the connectivity view as small nodes that share an edge with the object in focus. Clicking on these nodes allows users to navigate to the connected objects and to explore them in more detail.

#### 4.3.3 External Views

One essential feature of NeuroBlocks is the integration of a diverse set of external tools that support segmentation, proofreading, visualization, and analysis. Currently, we have integrated *Dojo*, a web-based proofreading application (see Figure 1), and *ConnectomeExplorer*, a scalable volume visualization and analysis application (see Figure 5).

*Dojo* supports fast volume navigation based on a 2D slice view and also offers a basic interactive volume rendering view for previewing the current segment in 3D. *Dojo* enables operations such as semi-automatic merging and splitting of segments by directly drawing and clicking on the segments in the slice view.

*ConnectomeExplorer* allows for high-quality volume rendering of terabyte-sized datasets. It can render the original data in combination with the segmentation data and users can interactively change parameters such as clipping planes, transfer functions, or enabling/disabling segments. Additionally, *ConnectomeExplorer* has a “diff” view for visual comparison between two segmentation states.

While working with external tools, their current state is periodically sent to NeuroBlocks and stored in a provenance database. These state snapshots are displayed as thumbnails in NeuroBlocks and enable users to reset their tool to any previous state to continue working from there. Furthermore, we have integrated the *Dojo* proofreading tool directly into the main view of NeuroBlocks. In this way, whenever the user selects a specific element, a 2D or 3D rendering of that segment or object is directly displayed in the sidebar of NeuroBlocks, without having to switch between different views or windows (see Figure 1).

#### 4.3.4 Management Views

Similarly to the object view, the *task view* provides the details for a specific task, its recent activity, and the people assigned to work on it. From this view, it is also possible to navigate to individual objects or segments that were modified in that task, and to explore the recent states of the used external tools (see Figure 1 top, second from right).

The *user view* is designed mainly for project managers and allows them to assign users to different groups with different default views and privileges. Furthermore, it offers an interface to *audit* users, the

tasks they have been working on, and the objects that they have modified. Other user groups can look at their profile to get summary statistics of their work. These summary statistics include a graph showing the number of contributions (i.e., segmentation changes) users made over time and a calendar view showing their activity over time.

## 5 INTERACTION DESIGN

The interaction features of NeuroBlocks were designed to support the domain-specific tasks discussed in Sec. 3.2 as intuitively as possible. NeuroBlocks supports multiple user roles that can have different privileges and default visualization views associated with them. By default, NeuroBlocks has three different user roles: *segmenters* and *proofreaders*, *senior segmenters*, and *project managers* — all with different default views and interaction possibilities. Project managers, for example, can create and assign tasks, audit segmentations, and assign users to different projects and tasks. Segmenters, on the other hand, can only modify segments based on their currently assigned tasks, and might need supervisor approval before their suggested changes are accepted into the current segmentation state.

### 5.1 Project Status Exploration

Exploring the overall status of the current project is mainly done in the segmentation state view and the timeline view. User interaction consists of filtering, sorting, and color-coding pixels in these views based on different criteria, but default configurations are provided to support specific tasks. For example, in a manual segmentation project, the project manager would see all segments, sorted by creation date and color-coded according to size. This enables users to see which segments have been added recently, and to follow the progress of the segmentation (i.e., segments growing in size as they are being traced through the volume by a segmenter).

Furthermore, users can get details on individual segments in the provided linked views, and explore the evolution of the segmentation over time with time sliders and brushes in the timeline.

### 5.2 Task Creation

Segmentation tasks are created by project managers during their project exploration process and often depend on previously-accomplished tasks. For example, manual segmentation projects typically start with completely segmenting a single object of interest and labeling its synapses (by marking the synapse locations in the image viewer). Objects of interest are selected by domain scientists based on their domain-specific analysis goal and are identified by the objects’ location, size, anatomy, or connectivity (e.g., a central, highly-connected dendrite). In the next step, new segmentation tasks are created for all connected structures sharing synapses. New tasks can be defined at any spatial position or segment, and proofreaders are automatically set to the specified starting position.

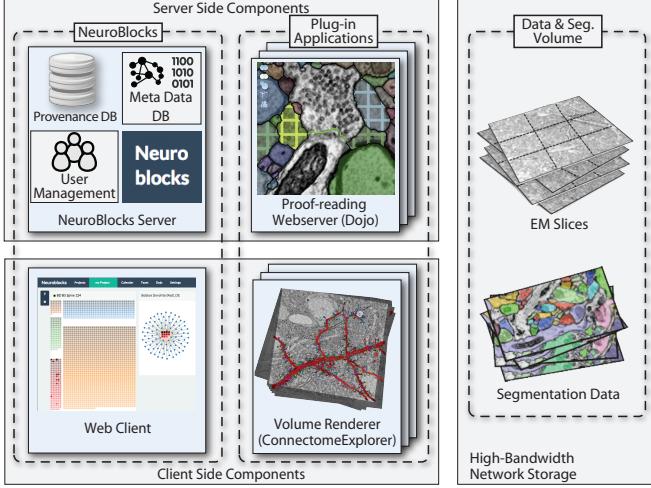
### 5.3 Task Completion

To minimize user interaction overhead, users who start working on a task are automatically transferred to the correct state and location in the integrated proofreading or segmentation tool. While they are working on their task, they can mark and report unclear areas in the data, and save additional comments. Their current progress is automatically saved in NeuroBlocks. Users can stop and interrupt their work at any time, and can later automatically resume at the exact same state, even when working on different computers. Once a task is marked as complete it will be submitted for approval to a project manager.

### 5.4 Task Approval and Audits

Project managers see a list of completed tasks and segmented objects that need approval, and can check the correctness of the data before they officially close a task or “lock” a segmented object. Locked objects are considered to be correctly segmented, and cannot be touched or modified anymore (unless explicitly re-opened).

Audits are a more formal evaluation of recent accomplished work to ensure that the project meets certain quality standards. NeuroBlocks supports audits for tasks, objects, segments and users. This helps in



**Fig. 6. NeuroBlocks system design.** The NeuroBlocks server integrates user management, handling of meta data, and a provenance database. External applications can use the NeuroBlocks plugin API to communicate with NeuroBlocks and access the original volume and segmentation data, which is stored on a shared filesystem.

identifying (junior) segmenters that need more training or do not have the required concentration for longer proofreading tasks.

## 6 SYSTEM DESIGN AND IMPLEMENTATION

The overall system design with the major components of NeuroBlocks is shown in Figure 6. NeuroBlocks is implemented as a web-based client/server application. The server integrates user management, handling of meta data, and can access the provenance database. The user only needs a web browser to connect to the server.

### 6.1 Data Management

The original EM volume and the current segmentation volume are stored in a multi-resolution format on a shared file system, and can either be accessed directly from there or via on-demand data streaming from a data server process [16]. To avoid having to store different snapshots of the entire segmentation volume over time, we only store the *changes* to the data in our provenance database, which allows us to roll back to previous versions without data loss. Currently, our provenance module is implemented as a MongoDB database. However, our modular design would allow us to easily plug in different provenance back-ends in the future, such as VisTrails [4], for example.

All segmentation meta data (e.g., textual labels, marked synapse locations, object attributes) are stored in a MongoDB database, which can easily be extended to incorporate new attributes or object types.

### 6.2 Multi-Tool Support

One of the main design decisions during the development of NeuroBlocks was to support the largest possible number of external tools that are or might be used by neuroscientists to segment, analyze, and visualize their data. NeuroBlocks offers a general API to allow external applications to plug into the system. The API has been designed in a way that makes it very easy to integrate external tools, with a minimum amount of required programming on the tool side. After a handshake protocol and user authentication, external applications send their states and meta data to NeuroBlocks and receive updates in turn.

Currently, we have integrated two external applications that exemplify how very heterogeneous tools can easily be integrated into NeuroBlocks: ConnectomeExplorer [6] is a GPU-based volume renderer that runs locally on the user's machine and is integrated into NeuroBlocks by using the general API. Dojo [17], on the other hand, is a web-based multi-user proofreading system for which we implemented a tighter coupling into NeuroBlocks. It still uses the general API for communication, but we display Dojo's views directly in NeuroBlocks, without using a separate browser window.

## 6.3 Implementation

NeuroBlocks is implemented in JavaScript and uses Meteor, an open source JavaScript framework for client/server applications. The interactive visualizations are implemented in D3. The provenance and data handling back-ends are built on MongoDB (a noSQL database). Communication with external tools is based on TCP sockets and web-sockets. To run NeuroBlocks, users only require a web browser, no installation is needed. To run Dojo with high-quality rendering settings, the web browser has to support WebGL. To run ConnectomeExplorer for interactive 3D visualization of the data, a standard Windows PC with at recent NVIDIA GPU is required.

## 7 CASE STUDIES AND DISCUSSION

We present two case studies that demonstrate how NeuroBlocks is useful in managing large-scale connectomics segmentation projects. We evaluated the case studies with our collaborating neuroscientists from the Harvard Center for Brain Science and the School of Medicine at Boston University. Throughout the project, we had regular meetings with them to discuss the goals and detailed tasks in NeuroBlocks. We performed informal as well as semi-structured interviews and used the *fly-on-the-wall* observation technique, to observe the scientists' typical workflow without direct participation or interference from our side.

### 7.1 Case Study 1: Managing a Segmentation Project

The first case study focused on a manually segmented data set of a 955 GB electron microscopy volume. Previously, the scientists had already sparsely segmented around 4,000 objects and labeled 943 synapses, by managing a team of summer interns and grad students. Since the first step of this segmentation had already been performed prior to the development of NeuroBlocks, we were only able to extract a small amount of provenance data from this initial segmentation state. However, for all modifications after the initial segmentation, we were able to fully employ NeuroBlocks and its provenance tracking capabilities. The initial segmentation of the EM volume was performed with VAST [5]. For each labeled synapse several attributes were extracted from the EM volume. In the later stages of the project, the scientists used Dojo [17] for segmentation and proofreading.

In this first case study, our target user group was a *segmentation project manager*—in our case a senior researcher. The main objective of the senior researcher was to quickly identify interesting areas of the segmentation, and to create tasks for his project members based on his exploration. The researcher started by exploring the main segmentation state view in NeuroBlocks, focusing on the current state of the segmentation (**T1 - Overview and detail visualization of current segmentation**). He was focusing on three main structures of interest: three large dendrites, and the structures that connect to them. The three dendrites and their synapses had already been segmented previously, but some of the structures connecting to these synapses had only been quickly marked and not segmented completely. The researcher quickly identified these structures by using the filtering capabilities of NeuroBlocks to only show the interesting subset of the data, and then sorted the pixel view according to segment size. Subsequently looking at the small segments that were connected to the dendrites in Dojo's integrated 3D view allowed him to identify new segmentation targets (**T3 - Manage fine-grained segmentation tasks**). The researcher then created and assigned new tasks, by filling out new task descriptions and specifying start positions for the follow-up segmentation in Dojo.

While the segmenters were working on the data, the manager regularly checked their progress, and discussed ambiguous areas and segmentations with them (**T2 - Track segmentation changes**). Once the users were done, the senior researcher double-checked their work, using ConnectomeExplorer's 3D and "diff" views, before approving the segmentation (**T4 - Audit users and segmentation**).

### 7.2 Case Study 2: Proofreading Automatic Segmentation

The second case study focused on an electron microscopy volume that was automatically segmented and required manual proofreading. The automatic segmentation was performed by the RhoANA pipeline [22],

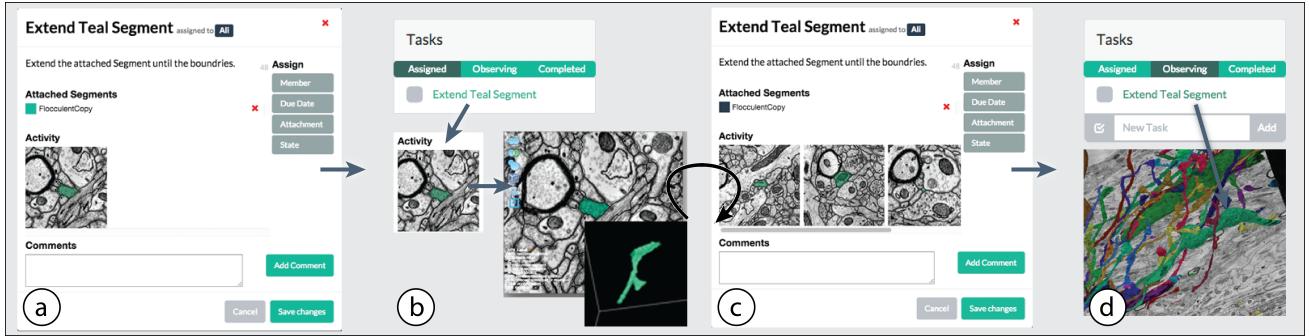


Fig. 7. *Case study 2: Proofreader workflow.* After task creation and assignment by a project manager (a), it is displayed in the segmenter's task list, which allows switching to the specified segment in the proofreading tool (b). Progress there is sent back to NeuroBlocks (c). After task completion, the project manager can double-check the new segmentation using the integrated volume renderer (d), and set the task to complete if satisfied.

resulting in a slightly over-segmented data set that required proofreading based on segment merging, and labeling of structures according to user-defined categories (e.g., axons, dendrites, spines).

For this case study, we focused on the *segmenter and proofreader* user group, and observed a user with relatively little proofreading experience. The complete workflow of this case study is shown in Figure 7. After logging into NeuroBlocks, the user could directly examine her assigned tasks and start working on one of them. Starting the proofreading process automatically sets the user to the specified position in Dojo (if a position was defined during task creation). During proofreading, all state changes (i.e., segment merges or splits) are automatically transferred to NeuroBlocks, which stores it in the provenance database (**T2 - Track segmentation changes**). During proofreading, the segmenter added several comments to her current task, to discuss ambiguous cell boundaries with more experienced segmenters (**T3 - Manage fine-grained segmentation tasks**). While waiting for answers, she suspended working on the current task and started other tasks that were assigned to her. After the ambiguity was resolved, she could directly continue with her first task where she originally left off, without losing the original tool state (**T5 - Seamless switching between different tools**). After checking her work in the integrated 3D view and setting the task status to complete, a segment approval notification was automatically triggered for the project manager.

### 7.3 Discussion

In this section we present the qualitative feedback we received from our collaborating scientists, and discuss limitations and as well as design lessons we learned during the development of NeuroBlocks.

#### 7.3.1 Qualitative Feedback

Qualitative feedback from our collaborators indicated three main advantages of NeuroBlocks that match our design goals:

First, it was a lot easier for neuroscientists to keep track of a project in NeuroBlocks than in their previous workflow. The distributed task assignment of NeuroBlocks was extremely helpful, especially when more than just one or two people are working on a project.

Second, NeuroBlocks was found to speed up and simplify the previous workflow. By seamlessly integrating external proofreading and visualization tools, users could easily switch from one tool to another and back again, without manual data importing and exporting, and without having to manually copy and paste meta data. Finally, users found NeuroBlocks most useful when the segments were categorized into the different neural object types defined by our collaborators' taxonomy (e.g., spines, dendrites, neurons) early in the process. The reason is that the segmentation state pixel view makes use of this categorization to display the segments sorted into the corresponding hierarchy. If the categorization is unknown, the visualization results in a flat view of the segments, which is still useful, but not as informative as the hierarchical view. Currently, the segment categorization is done manually during proofreading, but in the future the automatic segmentation pipeline could propose an initial object category for each segment.

#### 7.3.2 Lessons Learned

One of the crucial decisions during the design of NeuroBlocks was switching from a visualization and analysis-centric to a data-centric focus. This strategy allowed us to build a system that supports project management and tracking tasks much more intuitively by leveraging more abstract and high-level visualizations to get an overview of the current project state. This first design decision goes hand in hand with our next finding: By forgoing the typical graph-based provenance visualization, and providing a timeline-based provenance visualization triggered by users moving time sliders and brushes, we have created an intuitive way to explore the evolution of a project, even for users new to provenance systems. Tracking the overall progress of a project is a lot easier when one does not need to focus on individual changes or clusters, but can browse via smooth animations.

#### 7.3.3 Limitations

The main limitations of NeuroBlocks with respect to interoperability with external tools is that these tools need to use the NeuroBlocks API for transmitting and receiving state information. Even though the API is well-documented and flexible, this process usually requires access to the source code and recompilation of the external applications.

We have tested the scalability of our system by simulating provenance data for tens of thousands of proofreading steps in a teravoxel data set. Our multi-scale pixel view can potentially scale to thousands of segments per visual element and ensures on-screen visibility of all segments. However, as all multi-resolution and aggregation views, this comes at the cost of losing information on individual segments that are a part of a super-pixel. That is why we offer users the possibility to switch from the multi-scale pixel view with guaranteed visibility to a high-resolution pixel view with a scroll bar.

## 8 CONCLUSIONS AND FUTURE WORK

We have presented the design and implementation of NeuroBlocks, a system for visual tracking of large-scale segmentation and proofreading projects in the field of connectomics. NeuroBlocks enables scientists to track the current state and evolution of their on-going segmentation and proofreading project, and provides a scalable visualization that can handle thousands of segmented objects. We have incorporated a provenance management system for tracking the project evolution and auditing of segmentation and users, and support multiple users and multiple user groups with different default views, tasks and permissions. NeuroBlocks allows to seamlessly integrate external tools and handle their respective application states. We believe that by combining a high-level abstract pixel view of the entire segmentation data with highly-detailed spatial 2D and 3D views of the EM volume and its segmentation, we have created a powerful visual system that allows scientists to focus on the essential parts of their research, rather than on management, data handling, and interoperability issues.

In the future, we plan to extend NeuroBlocks into a complete virtual lab notebook that will allow scientists to not only track the progress of their segmentation, but also the progress of their visualization and analysis, and the discovery of new scientific insight. Another

promising research direction is the integration of dynamic and evolving guidelines for different types of science projects that will allow students with little experience to follow common templates.

## ACKNOWLEDGMENTS

This work was partially supported by King Abdullah University of Science and Technology and NSF grant OIA-1125087.

## REFERENCES

- [1] A. Al-Awami, J. Beyer, H. Strobelt, N. Kasthuri, J. Lichtman, H. Pfister, and M. Hadwiger. NeuroLines: A subway map metaphor for visualizing nanoscale neuronal connectivity. *IEEE Trans. on Visualization and Computer Graphics (Proc. IEEE InfoVis '14)*, 20(12):2369–2378, 2014.
- [2] A. Al-Naser, M. Rasheed, D. Irving, and J. Brooke. A visualization architecture for collaborative analytical and data provenance activities. In *Proc. Information Visualisation (IV '13)*, pages 253–262, 2013.
- [3] 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.
- [4] L. Bavoil, S. Callahan, P. Crossno, J. Freire, C. Scheidegger, C. Silva, and H. Vo. Vistrails: Enabling interactive multiple-view visualizations. In *Proc. IEEE Visualization '05*, pages 135–142, 2005.
- [5] D. Berger. VAST - Volume Annotation and Segmentation Tool. <https://software.rc.fas.harvard.edu/lichtman/vast/>, 2014. Accessed on 31/07/2015.
- [6] 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 Trans. on Visualization and Computer Graphics (Proc. IEEE SciVis '13)*, 19(12):2868–2877, 2013.
- [7] 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 and Applications*, 33(4):50–61, 2013.
- [8] D. Bock, W.-C. Lee, A. Kerlin, M. Andermann, G. Hood, A. Wetzel, S. Yurgenson, E. Soucy, H. S. Kim, and R. C. Reid. Network anatomy and in vivo physiology of visual cortical neurons. *Nature*, 471(7337):177–182, 2011.
- [9] S. Bruckner, V. Šoltészová, M. E. Gröller, J. Hladívka, K. Bühler, J. Yu, and B. Dickson. Braingazer - visual queries for neurobiology research. *IEEE Trans. on Visualization and Computer Graphics (Proc. IEEE Visualization '09)*, 15(6):1497–1504, 2009.
- [10] M. Bruls, K. Huizing, and J. van Wijk. Squarified treemaps. In *Proc. Joint EG / IEEE TCVG Symp. on Visualization '99*, pages 33–42, 1999.
- [11] R. Burns, W. G. Roncal, D. Kleissas, K. Lillianey, P. Manavalan, E. Perlman, D. R. Berger, D. D. Bock, K. Chung, L. Grosenick, N. Kasthuri, N. C. Weiler, K. Deisseroth, M. Kazhdan, J. Lichtman, R. C. Reid, S. J. Smith, A. S. Szalay, J. T. Vogelstein, and R. J. Vogelstein. The open connectome project data cluster: Scalable analysis and vision for high-throughput neuroscience. In *arXiv:1306.3543 [cs.DC]*, 2013.
- [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):e38011+, 2012.
- [13] J. Freire, D. Koop, E. Santos, and C. T. Silva. Provenance for computational tasks: A survey. *Comp. in Science and Engg.*, 10(3):11–21, 2008.
- [14] J. Freire and C. T. Silva. Making computations and publications reproducible with vistrails. *Comp. in Science and Engg.*, 14(4):18–25, 2012.
- [15] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011.
- [16] 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 Trans. on Visualization and Computer Graphics (Proc. IEEE SciVis '12)*, 18(12):2285–2294, 2012.
- [17] 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 Trans. on Visualization and Computer Graphics (Proc. IEEE SciVis '14)*, 20(12):2466–2475, 2014.
- [18] C. Halaschek-Wiener, J. Golbeck, A. Schain, B. Parsia, and J. Hendler. Annotation and provenance tracking in semantic web photo libraries. In *Proc. Intern. Prov. and Ann. Workshop (IPAW '06)*, pages 82–89, 2006.
- [19] B. Howe, P. Lawson, R. Bellinger, E. Anderson, E. Santos, J. Freire, C. Scheidegger, A. Baptista, and C. Silva. End-to-end eScience: Integrating workflow, query, visualization, and provenance at an ocean observatory. In *Proc. IEEE 4th Int. Conf. on eScience*, pages 127–134, 2008.
- [20] V. Jain, B. Bollmann, M. Richardson, D. Berger, M. Helmstädt, K. Briggman, W. Denk, J. Bowden, J. Mendenhall, W. Abraham, K. Harris, N. Kasthuri, K. Hayworth, R. Schalek, J. Tapia, J. Lichtman, and S. Seung. Boundary learning by optimization with topological constraints. In *Proc. IEEE CVPR '10*, pages 2488–2495, 2010.
- [21] E. Jurrus, S. Watanabe, R. J. Giuly, A. R. Paiva, M. H. Ellisman, E. M. Jorgensen, and T. Tasdizen. Semi-automated neuron boundary detection and nonbranching process segmentation in electron microscopy images. *Neuroinformatics*, 11(1):5–29, 2013.
- [22] V. Kayning, 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.
- [23] D. A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Trans. on Visualization and Computer Graphics*, 6(1):59–78, 2000.
- [24] J. S. Kim, M. J. Greene, A. Zlateski, K. Lee, M. Richardson, S. C. Turaga, M. Purcaro, M. Balkam, A. Robinson, B. F. Behabadi, M. Campos, W. Denk, H. S. Seung, and EyeWirers. Space-time wiring specificity supports direction selectivity in the retina. *Nature*, 509(7500):331–336, 2014.
- [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] 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, 2010.
- [27] J. W. Lichtman and W. Denk. The big and the small: Challenges of imaging the brain's circuits. *Science*, 334(6056):618–623, 2011.
- [28] F. Lindemann, K. Laukamp, A. H. Jacobs, and K. Hinrichs. Interactive comparative visualization of multimodal brain tumor segmentation data. In *Vision, Modeling & Visualization*, pages 105–112, 2013.
- [29] A. Rungta, B. Summa, D. Demir, P.-T. Bremer, and V. Pascucci. ManyVis: Multiple applications in an integrated visualization environment. *IEEE Trans. on Visualization and Computer Graphics (Proc. IEEE SciVis '13)*, 19(12):2878–2885, 2013.
- [30] 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.
- [31] R. Sadana, T. Major, A. Dove, and J. Stasko. Onset: A visualization technique for large-scale binary set data. *IEEE Trans. on Visualization and Computer Graphics (Proc. IEEE InfoVis '14)*, 20(12):1993–2002, 2014.
- [32] C. Scheidegger, H. Vo, D. Koop, J. Freire, and C. Silva. Querying and creating visualizations by analogy. *IEEE Trans. on Visualization and Computer Graphics (Proc. IEEE Vis '07)*, 13(6):1560–1567, 2007.
- [33] S. Seung. Reading the book of memory: Sparse sampling versus dense mapping of connectomes. *Neuron*, 62(1):17–29, 2009.
- [34] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1):92–99, 1992.
- [35] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. IEEE Symposium on Visual Languages (VL '96)*, pages 336–343, 1996.
- [36] Y. L. Simmhan, B. Plale, and D. Gannon. Karma2: Provenance management for data driven workflows. *International Journal of Web Services Research*, 5(2):1–22, 2008.
- [37] C. Sommer, C. Straehle, U. Kothe, and F. A. Hamprecht. Ilastik: Interactive learning and segmentation toolkit. In *IEEE Int. Symp. on Biomedical Imaging: From Nano to Macro*, pages 230–233, 2011.
- [38] 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, 2013.
- [39] A. Telea. Combining extended table lens and treemap techniques for visualizing tabular data. In *Proc. Eurographics/IEEE VGTC Conf. on Visualization (EuroVis'06)*, pages 51–58, 2006.
- [40] C. Wang, J. P. Reese, H. Zhang, J. Tao, Y. Gu, J. Ma, and R. J. Nemiroff. Similarity-based visualization of large image collections. *Information Visualization*, 14:183–203, 2013.