

Abstractocyte: A Visual Tool for Exploring Nanoscale Astroglial Cells

Haneen Mohammed, Ali K. Al-Awami, Johanna Beyer, Corrado Cali,
Pierre Magistretti, Hanspeter Pfister, and Markus Hadwiger

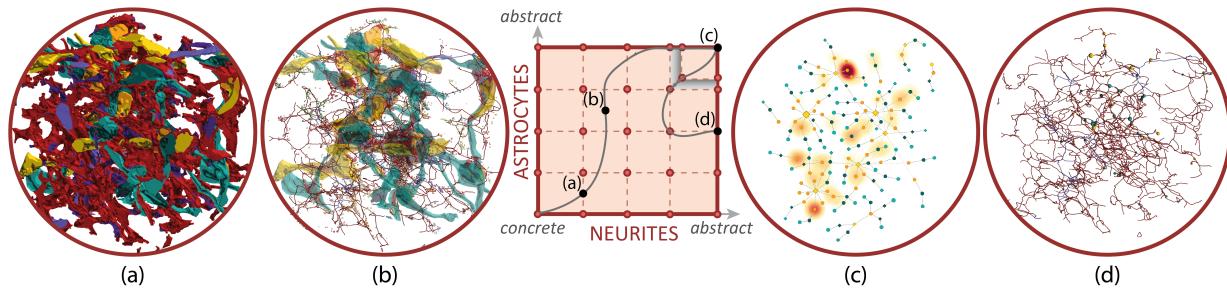


Fig. 1. **Visual exploration of astrocytes.** Our abstraction space panel (middle) allows users to intuitively navigate connectomics datasets containing both neurons and astrocytes at different levels of abstraction. The two axes of the abstraction space correspond to astrocytes and neurites, respectively, and range from concrete to abstract. The curve depicts the path taken during exploration, which can also be animated. (a) 3D mesh of astrocyte and neurites; (b) semi-transparent mesh and skeleton of neurites, simplified skeleton of astrocyte; (c) abstract node-link diagram of neurites, heat map of astrocyte proximity; (d) 3D nodes for neurites, skeleton of astrocyte.

Abstract—This paper presents *Abstractocyte*, a system for the visual analysis of astrocytes and their relation to neurons, in nanoscale volumes of brain tissue. Astrocytes are glial cells, i.e., non-neuronal cells that support neurons and the nervous system. The study of astrocytes has immense potential for understanding brain function. However, their complex and widely-branching structure requires high-resolution electron microscopy imaging and makes visualization and analysis challenging. Furthermore, the structure and function of astrocytes is very different from neurons, and therefore requires the development of new visualization and analysis tools. With *Abstractocyte*, biologists can explore the morphology of astrocytes using various visual abstraction levels, while simultaneously analyzing neighboring neurons and their connectivity. We define a novel, conceptual 2D *abstraction space* for jointly visualizing astrocytes and neurons. Neuroscientists can choose a specific joint visualization as a point in this space. Interactively moving this point allows them to smoothly transition between different abstraction levels in an intuitive manner. In contrast to simply switching between different visualizations, this preserves the visual context and correlations throughout the transition. Users can smoothly navigate from concrete, highly-detailed 3D views to simplified and abstracted 2D views. In addition to investigating astrocytes, neurons, and their relationships, we enable the interactive analysis of the distribution of glycogen, which is of high importance to neuroscientists. We describe the design of *Abstractocyte*, and present three case studies in which neuroscientists have successfully used our system to assess astrocytic coverage of synapses, glycogen distribution in relation to synapses, and astrocytic-mitochondria coverage.

Index Terms—Connectomics, Neuroscience, Data Abstraction, Interactive 3D Visualization

1 INTRODUCTION

In recent years, connectomics and the study of the detailed neural connectivity of the brain have become a main research focus in neuroscience. Scientists are looking at the mammalian brain at nanoscale resolution to reconstruct individual nerve cell connectivity, and to shed light on how the brain functions and develops. Interestingly, neurons (i.e., nerve cells) only make up about 50 percent of all cells in the mam-

malian brain [4]. The other half is made up by glial cells. These cells are typically located in the vicinity of neurons and provide supporting features for them. For example, they supply nutrients and oxygen, provide insulation, and form myelin. Astrocytes are a subtype of glial cells, which are crucial for regulating the transmission of electrical impulses within the brain, and also act as glycogen stores to supply neurons with energy. However, even though astrocytes are essential for the brain to function, they have so far received relatively little attention in connectomics research. One reason for this is probably that astrocytes are difficult to study. For example, the morphology of astrocytes is even more complex than that of neurons. Since they are supporting structures in the brain, they inherently connect to everything, making a detailed connectivity analysis complicated, if not impossible. Furthermore, while neurons have well-defined subparts (e.g., axons, dendrites, spines) that can be analyzed individually, astrocytes are much less structured, making a detailed analysis of individual parts difficult.

Most visualization and analysis tools for connectomics have focused on the synaptic connectivity of neurons, or on the overall structure and morphology of neurons in the brain [1, 2, 7, 9, 34]. To our knowledge, no visualization tool has tackled the analysis of astrocytes so far. When looking at astrocytes, neuroscientists are not only interested in

• Haneen Mohammed, Ali K. Al-Awami, Corrado Cali, Pierre Magistretti, and Markus Hadwiger are with King Abdullah University of Science and Technology (KAUST), Thuwal, 23955-6900, Saudi Arabia.
E-mail: {haneen.mohammed, ali.awami, corrado.cali, pierre.magistretti, markus.hadwiger}@kaust.edu.sa.

• Johanna Beyer and Hanspeter Pfister are with the John A. Paulson School of Engineering and Applied Sciences at Harvard University, Cambridge, MA, USA. E-mail: {jbeyer, pfister}@seas.harvard.edu.

Manuscript received 31 Mar. 2017; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxx

the astrocytes themselves, but also in how they interact with nearby neurons. Therefore, analysis tools should be able to highlight or indicate connections or closeness of astrocytes to neurons at different scales and levels of detail. For example, showing the entire astrocyte in a high-resolution 3D rendering will lead to clutter and obstructed views due to the highly complex morphology. However, a detailed 3D rendering might be the best choice when focusing on small details. Furthermore, while neurons also have a branching morphology at the microscopic level, at nanoscale resolution neurons are usually a lot less tangled than astrocytes. Therefore, during the analysis, it is crucial to be able to dynamically specify the level of detail that should be used in the current visualization of astrocytes and neurons, respectively.

In this paper, we present *Abstractocyte*, a novel tool for the visual exploration of astrocytes and neurons. We tackle the inherent data complexity of nanoscale brain tissue by supporting a multitude of different visualizations that display the data at different levels of abstraction. Visualizations range from detailed 3D surface renderings of segmented structures to simplified abstractions, e.g., skeletons and graphs. We allow the abstraction of both entities (i.e., astrocytes and neurons) independently of each other. This is necessary because biologists need to be able to explore both types of cells in combination, but often at different levels of detail, e.g., a detailed mesh of a neurite with a simplified skeleton of the astrocyte. We achieve this by presenting a novel interaction interface that we refer to as an *abstraction space* (see Fig. 1). This conceptual space allows users to intuitively explore the space of possible abstractions. Furthermore, we support the analysis of the glycogen distribution in astrocytes. Our collaborators are currently investigating whether the glycogen distribution within an astrocyte is correlated with factors such as the synapse strength of nearby neurons. Therefore, we cluster and display the glycogen distribution, and map it to structures of interest, e.g., to nearby synapses or mitochondria.

The main contribution of our work is the design and implementation of Abstractocyte, as a novel tool for the interactive visualization of nanoscale astroglial cell morphology. We introduce an abstraction metaphor that allows users to explore astrocytes in relation to their surrounding neurons at different levels of visual abstraction, for astrocytes and neurons, respectively and independently. Abstractocyte also supports the visual analysis of the detailed glycogen distribution and neurite proximity of astrocytes. Furthermore, we demonstrate an intuitive and flexible user interface for exploring the 2D visual abstraction space, which enables interactive and consistent transitioning between 2D and 3D visualizations. We illustrate the usefulness of Abstractocyte in practice via three case studies from our collaborators in neuroscience.

2 RELATED WORK

Neuroscience and Connectomics. Connectomics aims to reconstruct the detailed connectivity diagram of neurons in the brain using high-resolution and high-throughput imaging [27]. The main goal is to understand the relation between structure and function of the brain [31].

Visualization for Connectomics. Pfister et al. [30] give a survey of different techniques for macro-, meso-, and microscale connectivity visualization for connectomics. An overview of different frameworks for visualizing the human connectome is given by Margulies et al. [28].

Most previous visualization tools for connectomics focus on high-level connectivity visualization between entire brain regions. In these frameworks actual cell morphology is typically abstracted, and the focus lies on depicting the connectivity between regions or a network of neurons. For example, NeuroMap [34] uses circuit wiring diagrams to represent possible connections between neurons. Other tools for high-level connectivity visualization make use of 2D projections for 3D tractography data [21], or use matrix visualizations for showing connectivity information [29]. NodeTrix [38] is a system for the block-wise (i.e., region of interest based) comparison of human brain networks. Several toolkits have been developed for the visual analysis of brain networks [17, 26]. All of these methods, however, focus on high-level or macro-scale connectivity information between entire brain regions. Our collaborators are interested in nanoscale brain morphology and connectivity at the level of individual synapses. On a cell level, individual neurons can be represented as nodes in a connectivity graph,

with synapses being the actual links between nodes. Examples for this are shown in the Viking Viewer [3] and ConnectomeExplorer [7]. Several tools allow users to define interactive queries for analyzing brain connectivity data, such as BrainGazer [9] or ConnectomeExplorer [7]. None of these tools, however, support the visualization of astrocytes.

Astrocyte analysis and visualization. Braumann et al. [8] have introduced an image processing chain to generate structural descriptions of astrocytes based on graphs. They use confocal laser scanning data and therefore work on the macro level. Suwannarat et al. [36] use confocal microscopy and probabilistic segmentation to visualize astrocytes of the retina. They use a glyph based visualization where individual glyphs are scaled according to the uncertainty of the segmentation. More recently, Cali et al. [10] have used a CAVE for visualizing high-resolution neuronal EM datasets. They also segment and render glycogen granules using Blender and NeuroMorph [22]. None of these methods, however, support an interactive exploration of the data in 3D with dynamically changing levels of detail or abstraction.

Visual abstractions for biological structures. To analyze the connectivity of individual neurons, Neurolines [2] introduces a visual abstraction of neurites that reduces clutter and converts the 3D structure of neurons into a 2D visualization. A different 2D abstraction of 3D data was introduced by Borkin et al. [7] for the visualization of artery trees for the diagnosis of heart disease. More generally, CPR (curved planar reformation) [23] allows 3D tubular structures such as vessels to be displayed in a 2D visualization with minimal loss of information. All these methods focus on anatomical structures that can be represented as trees or a collection of trees. However, the nanoscale morphological structure of astrocytes is more difficult, and typically contains loops. More recently, Smit et al. [32] used a topological similarity measure and an abstract graph representation to visualize anatomical variations in complex branching structures for educational purposes. Sorger et al. [35] used visual abstractions for molecular visualization. They propose abstraction transforms for smooth transitions between different visual models (e.g., different developmental phases of a virus).

2D selection/input panels. Kniss et al. [23] introduced multidimensional transfer functions for GPU volume rendering. They also proposed a 2D classification widget or panel that allows users to interactively specify the transfer function mapping in 2D. A more general concept of interactive UI sliders was recently presented by Tsandilas et al. [37]. They propose a sketching interface for exploring multi-dimensional datasets. Abstractocyte uses a panel that allows users to interactively explore the abstraction space of possible visualizations. Users can freely navigate within the 2D panel, and directly see their current position in the abstraction space visualized in the main view.

Animated transitions between visual representations. Heer and Robertson [19] have shown that animated transitions between different statistical charts improves graphical perception. Elmqvist et al. [13] introduce animated transitions for exploring multidimensional data with scatterplots, using animated rotations in 3D space. More recently, Guillaume et al. [18] have presented four techniques for animating changes in radial tree visualizations. Regarding volumetric data, Basch [5] has investigated animated transitions across multiple dimensions, for example, transitions between volume rendering and histogram views. Zwan et al. [39] propose a technique for navigating between nested abstraction levels in volumes, using image-based halos and magic lenses. They demonstrate their technique on a fluid flow visualization. In this paper, we focus on smooth transitions between different abstraction levels of volumetric biomedical data. The user can seamlessly switch from highly detailed surface rendering to simplified views such as 3D skeletons or 2D node-link diagrams.

Network layouts. Connectomics focuses on the connectivity of neurons. Therefore, most visualization tools for connectomics include network and graph visualization components. Node-link diagrams are the most common form for representing the connectivity between brain regions and/or individual neurons. If the inherent 3D structure of the anatomical features is still available, connectivity can be shown on top of the original spatial position [9, 29]. In more abstract connectivity visualizations, on the other hand, the spatial location of nodes is completely independent of their original spatial positions [7, 34]. Hybrid

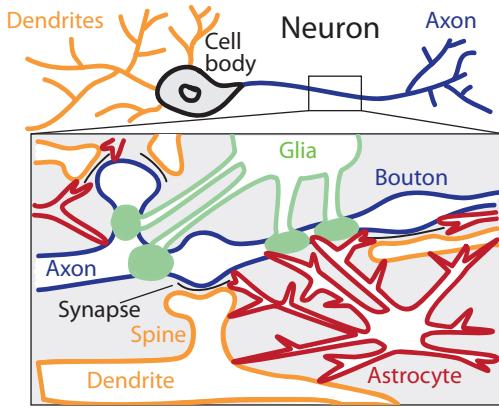


Fig. 2. **Structure of neurons and glia.** Neurons transmit signals from axons to dendrites via synapses. Glia are cells that provide supporting functions for the neural system. They are highly interconnected with neurons, but do not transmit nerve impulses. Astrocytes are a type of glial cell that are especially important for synapse formation and function.

approaches like Neurolines [2] retain some of the spatial information in their 2D graph layout. In Abstractocyte we use force-directed graph layouts to compute the abstract 2D views of our data. Kobourov [24] gives a good introduction to force-directed graph drawing algorithms. Fruchterman and Reingold [16] originally proposed spring forces where there are repulsive forces between all nodes, but also attractive forces between nodes that are adjacent. Later, Frick et al. [15] added new heuristics to the original spring forces approach by Fruchterman and Reingold. In Abstractocyte we have added additional heuristics to our graph layouting algorithm that are specific to our domain (Sec. 6.1).

3 BIOLOGICAL BACKGROUND

The average human brain contains around 100 billion *neurons*. Neurons are highly interconnected, and transmit nerve impulses by forming *synapses* with each other. A single neuron typically consists of a cell body, multiple dendrites, and a single axon. See Fig. 2. *Dendrites* are tree-like structures that receive signals from other neurons, while *axons* are long tubular structures that extend from the cell body to forward signals to other neurons. The term *neurite* can be used to describe either an axon or a dendrite. Synapses are made up by a *bouton* on the side of the axon, the synaptic cleft, and the post-synaptic terminal on the receiving dendrite. Dendrites can have synapses either directly on their trunk, or on small protrusions called *dendritic spines*.

While neurons make up around 50 percent of the cells in the mammalian brain, the other half is made up by *glial cells* [4]. Glial cells are non-neuronal cells that typically surround and insulate nerve cells, and provide supporting features for neurons [20]. An *astrocyte* is a type of glia that is star-shaped (Fig. 2). It is mainly responsible for providing nutrients, balancing the amount of neural transmitters, and maintaining the blood-brain barrier. Astrocytes make up 20 to 40 percent of all glia, and are important for synapse formation and function [12]. They also act as glycogen stores to supply neurons with energy. Our collaborators are interested in the distribution of glycogen granules within astrocytes, and whether there is a correlation between glycogen granules and biological features such as synapse strength. They also want to investigate how the mitochondria distribution in astrocytes influences cell behavior.

3.1 Neuroscience Workflow

Our collaborators start their workflow by scanning a mouse brain using electron microscopy, before they register the individual scanned slices to form a consistent 3D volume. The size of an imaged tissue block is typically five microns in each dimension, which is imaged at a resolution of $5 \times 5 \times 15$ nm per voxel. This results in volumes of about $1024 \times 1024 \times 450$ voxels in size. These volumes are then typically segmented using the semi-automatic methods provided by Ilastik [33] and TrakEM2 [11]. Segmentations are created for all astrocytes, axons, dendrites, synapses, spines, boutons, glycogen, and mitochondria.

4 ABSTRACTOCYTE DESIGN

The idea of Abstractocyte originated in meetings with our collaborators, where they described their problems in analyzing and exploring data acquired for investigating astrocytes in an intuitive manner. Biological structures in brain tissue at nanometer resolution tend to have an incredibly complex morphology that easily leads to clutter or occlusion in 3D visualizations. However, not all analysis and exploration tasks require the data to be shown with full detail. It can often be better to show different structures at different levels of detail or abstraction. Therefore, the main goal of our work was to provide neuroscientists with the framework necessary to study brain cells, including nerve cells as well as glial cells, in an interactive and intuitive manner, at the visual abstraction level that is best suited for each individual task. To achieve this, users have to be able to seamlessly navigate between different visual abstraction levels, to reduce complexity while preserving essential features related to morphology, connectivity, and proximity.

4.1 Domain Goals and Tasks

The design of Abstractocyte supports the high-level domain goals of neuroscientists to (a) explore the detailed **morphology of astrocytes**; (b) analyze **proximity patterns between astrocytes and neurites** (also called astrocytic coverage); and (c) analyze **energy (or glycogen) distribution patterns**. These overall goals can be broken down into the following specific domain-oriented tasks:

T1 – Explore the astrocytic coverage of neurites and synapses. Our collaborators are interested in how much of a structure (e.g., a synapse) is covered or touched by an astrocyte. Synaptic coverage is plastic—it is different depending on the brain area and the activity of the synapses. Our collaborators want to investigate whether there is a causal relationship between coverage, synaptic plasticity, and synapse maturation.

T2 – Synapse glycogen analysis. Currently, there is still a debate in neuroscience whether boutons or spines are energetically more expensive during synaptic transmission. Therefore, our collaborators want to investigate how and to which extent glycogen granules (which are a precursor of lactate, and therefore an energy store located in astrocytes) are related to boutons and spines. Looking at synapses and their proximity to glycogen clusters will be a first step in that direction.

T3 – Mitochondria glycogen analysis. Our domain experts are investigating whether the presence of mitochondria in neurons can help in the interpretation of the spatial relationship of glycogen granules and synaptic elements. For example, dendrites that have mitochondria along their shaft might accumulate glycogen granules around excitatory shaft synapses, from which spines generate. This would suggest an involvement of astrocytic energy in the formation of dendritic spines.

T4 – Astrocytic mitochondria coverage of synapses. Mitochondria are the powerhouses of the cell. In astrocytes, mitochondria can have other functions, such as modulating intracellular calcium signaling. This has been shown to be related to synaptic activity, and can be modulated by mitochondria. Therefore, our collaborators want to explore and compare the locations of astrocytic mitochondria and synapses.

T5 – Astrocytic mitochondria coverage of their parent astrocyte. Mitochondria locations within astrocytes are not spread out, but appear preferentially in microdomains. Our collaborators want to explore whether this is related to the presence of synapses and, therefore, want to study the intracellular distribution of mitochondria within astrocytes.

5 ABSTRACTION SPACE

The necessity of supporting different levels of visual abstraction comes from the observation that the optimal visual representation of neurites and astrocytes changes depending on the current user objective. For example, when neuroscientists want to explore the neuronal connectivity, they prefer to use an abstract node-link diagram for neurites rather than a highly detailed mesh (Fig. 1c). On the other hand, when they want to analyze how close neurites are to a specific astrocyte, they would want to look at the neurite in more detail, and either color-code the proximity of the astrocyte onto the neurite's surface, or show the astrocyte as a morphological 3D skeleton (Fig. 1b). In order to be able to do this,

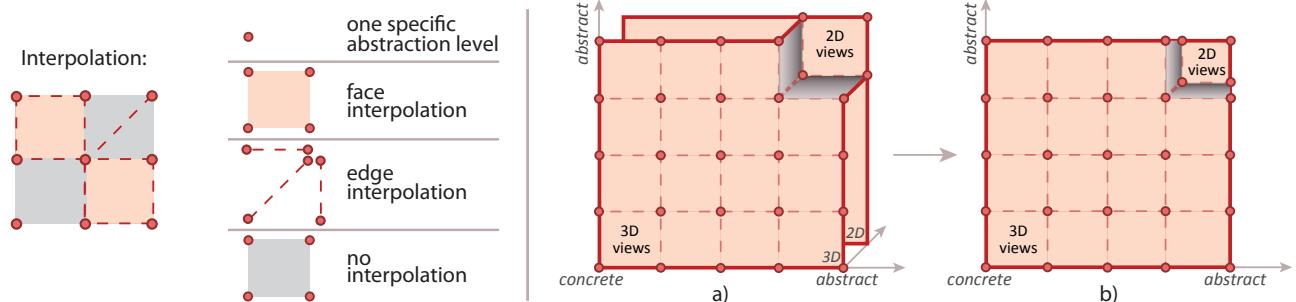


Fig. 3. Visual abstraction space. Left: We define three interpolation methods to support smoothly navigating the 2D abstraction space: face interpolation, edge interpolation, and no interpolation. When choosing visual abstractions, our system restricts user input depending on the supported modes of interpolation between specific abstraction levels. Right: (a) Our abstraction space supports transitions between 2D and 3D views, which are conceptually arranged on separate layers. (b) In the user interface, we depict these layers in simplified form, using a diagonal edge between them. Edge interpolation for the transition from 3D to 2D, or vice versa, enforces a simultaneous change of the abstraction level for both neurites and astrocytes. All other abstraction levels in Abstractocyte currently use face interpolation, where both abstraction levels can change independently.

scientists need to be able to change the level of abstraction for different structures on the fly, as their analysis progresses.

In Abstractocyte, we introduce a 2D visual abstraction space that allows us to abstract and simplify the visual representation of structures along two orthogonal, conceptual “axes” (see Fig. 1, center). The first axis represents astrocytes, and allows users to seamlessly switch the rendering of astrocytes from detailed (e.g., high-resolution 3D meshes) to abstract (e.g., a 2D node-link diagram of the astrocyte’s morphology). The second axis represents neurites, and allows interactively changing the visual representation of neurites. Fig. 1 shows this abstraction space and the visual abstractions at four selected points within that space.

5.1 Abstraction Space Elements

The abstraction space is spanned by the two axes that represent neurites and astrocytes, respectively, going from concrete, at the origin, to abstract. Supported specific abstraction levels along each axis are represented by points or nodes along that axis. Combined, the abstraction levels of both axes create a 2D grid of specific abstraction levels. Fig. 3 (left) shows the different elements in the abstraction space:

Nodes: Each node represents a specific combination of a neurite and an astrocyte abstraction level (e.g., a 3D mesh for one structure, and a 3D skeleton for the other one). At the exact position of a node, no interpolation between abstraction levels is necessary. The representations for neurites and astrocytes are combined directly during rendering.

Edges: We support edges between two nodes that are either axis-aligned, or at a 45-degree angle. In the former case, the edge represents the linear interpolation between two abstraction levels along the corresponding axis. In the latter case, the edge represents the linear interpolation between abstraction levels for both axes at the same time. A 45-degree edge ensures that the abstraction levels of both axes are changed simultaneously. This is necessary in certain cases to ensure consistency of the visualization, for example, when moving from a 3D representation to a 2D representation. Usually, a visualization where one of the axes has a 2D, and the other one a 3D representation is not meaningful. Therefore, we prevent this by restricting user input to a 45-degree edge, so that the dimensionalities for both axes are linked.

Faces: Four neighboring nodes in the abstraction space span an area. Selecting a point inside that area will automatically compute an interpolated abstraction level. We compute the interpolated value along each axis separately, and use these values to create the visual abstractions for neurites and astrocytes, respectively. The combination of neurite and astrocyte abstraction levels is done automatically during rendering.

Interpolation: For every node and combination of nodes we specify what type of interpolation it supports depending on the underlying abstraction. By default, we allow full face-based interpolation between nodes. However, we can restrict user navigation inside the abstraction space to edges, i.e., the abstraction level can only be modified along edges, or even disallow interpolation altogether. The different interpolation options that we support are illustrated in Fig. 3 (left).

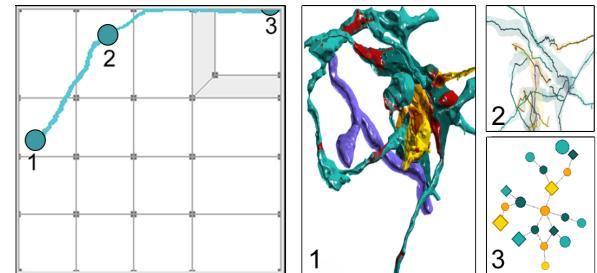


Fig. 4. Abstraction space GUI panel. (Left) Users can draw and save paths in the abstraction space for a continuous animation through the abstraction levels along them. (1, 2, 3) Paths can be replayed or navigated via a slider to revisit specific levels of abstraction. (1) Detailed surface rendering of a spine (orange) connected to four different boutons (teal), and close to mitochondria (blue); astrocyte proximity is visualized using splatting and color coding (red) on the neurite surfaces; (2) Neurites as skeleton; (3) 2D node-link diagram showing the connectivity of neurites.

Switching dimensionality of the visual abstraction: Conceptually, the abstraction space allows switching between arbitrary visual abstraction levels of the underlying data. However, when switching the dimensionality of the underlying data representation and visualization, some extra considerations need to be taken into account. First, we want to visually display this switch in dimensionality in the abstraction space. For this, we have extended the abstraction space to the concept of layers (Fig. 3a). In the top right corner, the abstraction switches from a 3D representation to a 2D representation. Therefore, we indicate the transition from a 3D to the layer below (2D). This also allows us to restrict user navigation to edge interpolation in the area where the abstraction moves from one dimensionality to another. Using edge interpolation makes sure that the resulting visualization is always consistent (i.e., valid) and that it does not mix arbitrary 2D and 3D abstraction levels. After feedback from our collaborators, we have further adjusted the user interface to the simplified layer representation shown in Fig. 3b.

5.2 Abstraction Space GUI Panel

Our 2D user interface panel for navigation inside the abstraction space is shown in Fig. 4. Users can freely move the mouse pointer in order to select or change abstraction levels for neurites and astrocytes, respectively. As the user moves between different locations in the panel, we smoothly interpolate between abstractions, depending on the current position and its neighborhood. However, at the same time our system restricts user interaction to those areas in the panel where valid abstractions have been defined (invalid areas are grayed out).

We also allow users to draw entire paths in the GUI panel (Fig. 4, left), which allows re-using and replaying a sequence of abstractions. This is very useful for repeating standard analysis tasks, where users can pre-define which levels of abstraction they need at what point, as well as for presentation and teaching purposes, to reproduce an analysis.

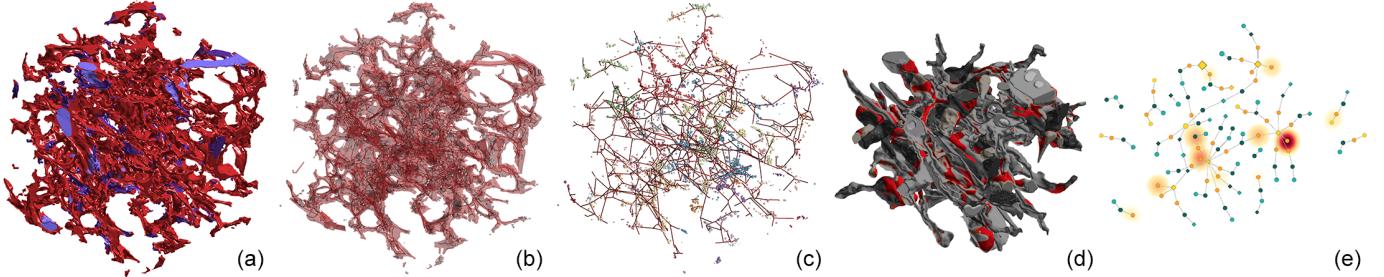


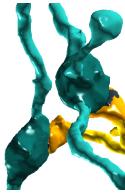
Fig. 5. Astrocyte abstraction levels. Astrocytes displayed at different levels of abstraction. (a) Highly detailed 3D mesh (L0); (b) Transitional phase, where the 3D mesh contracts and fades to a skeleton (L1.5); (c) Simplified skeleton together with clustered glycogen granules (L3); (d) Astrocyte shown as proximity map on top of neurites (L4, 3D); (e) Astrocytic coverage shown in 2D node-link diagram of neurites as a heatmap (L4, 2D).

6 VISUAL ABSTRACTIONS FOR BRAIN CELLS

The main goal for visualizing neurites is to be able to identify important regions and sub-structures, such as the sites of synapses (i.e., boutons and spines), and their relation to astrocytes. The main goal for visualizing astrocytes is to allow scientists to explore their proximity to neurites, while at the same time showing important information about the astrocyte (e.g., glycogen granules or mitochondria). In collaboration with our domain scientists, and keeping in mind the overall domain goals, we have identified the following visual abstractions (Fig. 5) :

Abstraction Level 0 (L0):

We use surface rendering of the high-resolution 3D mesh of the segmented data (i.e., neurites, astrocytes, and all their segmented sub-structures). This is the most detailed visualization, and allows users to see the data in its original form and highest detail. Structures are color-coded by type, to quickly provide an overview of what type of structures are present.



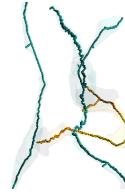
Abstraction Level 1 (L1):

We use surface rendering with non-photorealistic shading. We have chosen toon shading to retain the overall 3D impression, while abstracting minor bumps or edges from neurites or astrocytes, respectively. This is useful when looking at astrocytes, as the high-resolution mesh is often too detailed and distracts from overall morphological patterns.



Abstraction Level 2 (L2):

We render the 3D skeletons of neurites or astrocytes, respectively. Skeletons reduce clutter in the visualization, while still maintaining important spatial cues. The high-resolution mesh of an astrocyte/neurite often occludes synapse details, while a skeleton is a sparser visual encoding that still allows seeing morphological (or topological) features of the structure. By pre-computing a mapping of each mesh to its skeleton, we can smoothly interpolate between the two representations.



Abstraction Level 3 (L3):

We present neurites/astrocytes as simplified 3D skeletons, where branches have been straightened. This abstraction can be used to highlight 3D connectivity. Anatomical details are removed, while the user still gets an impression of the location, orientation, and branching pattern of a neurite. Our collaborators use this mode when they focus on branching patterns and properties.



Abstraction Level 4 (L4):

In this abstraction level, we use different representations for neurites and astrocytes, respectively. We further distinguish between abstraction level 4 in 2D or 3D views, and adjust the representation accordingly. Neurites are displayed as nodes. In the 2D view, we show a full node-link diagram where neurites are represented as nodes, while connections between them (i.e., synapses) are shown as links between nodes. In the 3D view, neurites are also shown as nodes. However, we



omit the links between them. The reason for this is that we typically have a high number of links, and 3D node-link diagrams get cluttered easily. This is the most abstract neurite visualization, and is mainly used for exploring synapses and connectivity. Optionally, we can map the volume of a neurite to its node size, to give an indication of its size.

The visual encoding of astrocytes in L4 is done differently. We do not render astrocytes as their own primitives anymore, but project them onto the closest neurite’s surface. This allows us to show the distribution or coverage of astrocytes on neurons and their substructures. This is especially useful to judge how close an astrocyte is to a given neurite.

Combining different visual abstractions: The power of Abstractocyte comes from the ability to combine different visual abstraction levels for neurites and astrocytes. This allows users to pick the most abstract level for their task that still provides them with enough details to answer their current questions. Furthermore, we can display additional information on a segment’s surface mesh, skeleton, or connectivity graph, by using color mapping and density fields (Sec. 7.2).

6.1 Transitions between Abstraction Levels

While users navigate the abstraction space (by moving the current mouse position), all views are continuously updated to reflect the corresponding levels of abstraction. Positions that are between specific levels of abstraction are interpolated using the interpolation methods described above. For transitioning between two abstraction levels that both employ a 3D visualization (i.e., meshes, skeletons, simplified skeletons), we linearly interpolate between the two 3D abstractions. In particular, we interpolate shading for transitioning to non-photorealistic rendering, and we interpolate vertex positions and opacity for transitioning between a mesh, a skeleton, or a simplified skeleton.

3D to 2D transitions. Transitioning from a 3D representation to a 2D representation is handled differently. First, we restrict user input to edge interpolation, so that astrocytes and neurites are switching dimensionality at the same rate. Next, we use a customized, force-directed layout algorithm to convert 3D graphs into valid 2D representations [24]. Neighboring nodes in a force-directed layout repel each other unless they are connected, in which case they attract each other. We have added three new rules to the algorithm to make sure we achieve an optimized layout. First, nodes are attracted to the center of the canvas. This ensures that the graph does not expand rapidly. Second, nodes are attracted to their original 2D screen position, i.e., their position before starting the force-directed layout. Third, nodes are constrained to stay within a certain radius of their original 2D screen position. Together with the previous rule, this keeps the graph similar to the original 3D skeleton, retaining its morphological form while resolving occlusions, and facilitating smooth transitions.

Additionally, when transitioning from a 3D to a 2D view, we adjust the weights described above for the force-directed graph algorithm according to the current interpolation position, to achieve the desired 3D to 2D graph layout transition. The individual forces are accumulated and applied to the nodes to update their positions accordingly. This is repeated until the graph is in a steady state, or the user changes the abstraction level or the viewport.

To optimize the performance of the force-directed graph layouting, we use 2D spatial hashing to find neighboring nodes efficiently.

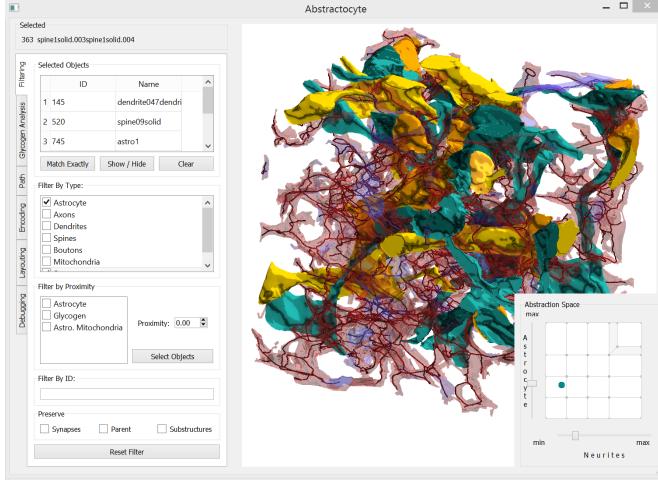


Fig. 6. Abstractocyte user interface. The filtering and analysis panel is displayed on the left. It allows users to interactively filter, select, and sort the data. The abstraction space panel is shown at the bottom right.

7 VISUAL ANALYSIS

The main goal of the different visual abstractions introduced above is to support scientists in the analysis of their data. In accordance with our three main domain goals described in Sec. 4.1, we support the three different exploration and analysis scenarios described below.

7.1 Visual Exploration of Morphology and Topology

After acquisition and segmentation of a new dataset, the first objective of our collaborators is to get a good mental image of the data. Therefore, exploring the morphology and topology of astrocytes as well as neurites is typically their first objective. Allowing users to look at their data at different abstraction levels, with smooth transitions between levels, helps them to identify morphological features for cells of interest. For example, when using a skeleton representation for an astrocyte it is easier to follow branches, and see how they interact with their surroundings. On the other hand, a simplified 2D graph representation allows users to quickly see the connectivity between structures, without having to manually trace and follow structures in 3D. In general, there is no one specific abstraction level that allows users to see all interesting features of a cell of interest. Only frequent transitioning between different levels of abstraction enables scientists to gain a detailed understanding of their data. Being able to freely navigate inside the abstraction space is particularly important for tasks **T1** and **T5** (Sec. 4.1), to explore the overall relations of neurites, astrocytes, and mitochondria.

In addition to providing visual abstractions, we allow users to filter their data on-the-fly, to exclude unimportant structures from analysis and visualization. Data can be filtered by manual selection directly in the visualization, or by using a user interface filter panel. This panel supports a combination of filters, based on object type, object ID, and additional parameters, such as connectivity, parents, and substructures (Fig. 6). Filtered-out objects are removed from all subsequent visualization and analysis steps. This is important for tasks **T1** to **T5**.

7.2 Visual Proximity Analysis

One of the most important tasks when analyzing brain cells and networks is spatial proximity analysis. In Abstractocyte, we can visualize the proximity of a cell structure *A* to another structure *B*, by conceptually “splattting” structure *A* onto the visualization of structure *B*. See Fig. 7b for an example. To be able to do this interactively, we first create a 3D binary segmentation texture for structure *A*, setting voxels belonging to *A* to one, all other voxels to zero. If available, we create this texture directly from voxel segmentation data. If not, we voxelize the 3D segmentation mesh of structure *A*. In the next step, we use this texture for distance queries while rendering structure *B*. Specifically, at every vertex position of *B*, we take a number of samples in the vicinity of that vertex in the previously generated 3D texture of structure *A* to

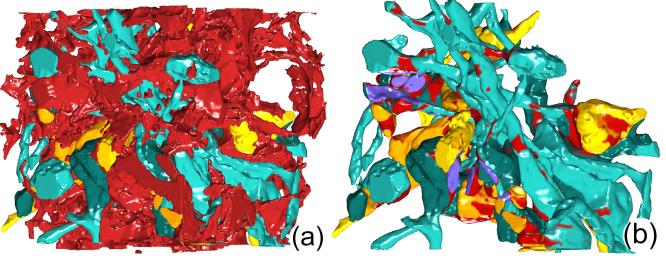


Fig. 7. Visual proximity analysis. (a) The astrocyte (red) occludes the neurites (cyan and yellow), making it hard to see where the astrocyte touches or is in close proximity to the neurites; (b) The proximity of the astrocyte is visualized on the neurite's surface using splatting and color mapping (the red areas on the neurite surface). The proximity of astrocytic mitochondria is visualized in the same way (purple areas).

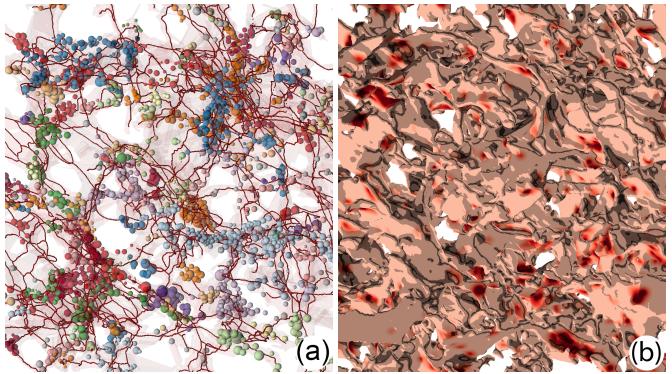


Fig. 8. Glycogen analysis and visualization. (a) Clustered glycogen granules depicted in combination with the astrocyte's skeleton; (b) Glycogen density mapped onto the toon-shaded 3D mesh of an astrocyte.

estimate the distance between the two structures. We usually weight the resulting distance with a Gaussian curve to obtain a smooth fall-off. Using this basic approach, we visualize spatial proximity in two different ways, depending on the dimensionality of the visualization:

Proximity in 3D. In the case where structure *B* is visualized as a 3D mesh or a 3D skeleton, we color-code the surface of the mesh or the skeleton using the technique above. When color-mapping values onto a skeleton's surface, we can either directly use the skeleton coordinates to query the distance values, or we use the corresponding mesh vertex coordinates. Fig. 5d and Fig. 7b depict examples of this approach.

Proximity in 2D. For more abstract 2D representations, e.g., a node-link diagram of neurites, we can visualize the distance values as a heat map on top of the network diagram. For this, we further splat all neurite nodes into a 2D texture, and attenuate a node's intensity by its distance to the astrocyte. The resulting 2D texture can directly be displayed as a heat map on top of the graph. This approach is depicted in Fig. 5e.

Filtering. Abstractocyte also allows on-the-fly filtering based on proximity and coverage. This allows us to restrict the visualization to structures within a certain minimum and/or maximum distance to a selected structure. For example, this allows looking only at synapses with a minimum amount of astrocytic coverage. Visual proximity analysis supports tasks **T1** (to analyze astrocytic coverage of neurites), **T2** and **T3** (to analyze glycogen), and **T4** (to analyze mitochondria).

7.3 Visual Analysis of Glycogen Granules

A major objective of Abstractocyte is to allow scientists to analyze the glycogen distribution in astrocytes, and to explore the relationship between glycogen cluster locations and synapses. Glycogen granules are stores of energy in the form of glucose. Brain energy consumption is of major interest in neuroscience, where the goal is to find the mechanisms of how energy is being regulated, distributed, and absorbed by neuronal cells. Astrocytes hold and regulate a lot of that energy in the form of glycogen. To tackle these questions, our collaborators identify and mark the spatial locations of biological granules (in our case glycogen

granules) in the microscopy data. Glycogen tends to appear in clusters in different locations inside the astrocyte. We have implemented on-the-fly clustering of granules using the DBSCAN (density-based spatial clustering of applications with noise) algorithm [14].

We support two different ways of glycogen visualization: First, we can visualize individual granules directly by rendering them as small spheres in 3D (Fig. 8a). We can also color-code granules based on their cluster membership. Second, we can compute granule density maps that we can then use in our 3D and 2D visualizations (Fig. 8b). Computing the granule density map is done similarly to computing the proximity between structures (Sec. 7.2), and allows us to map granules to their closest subcellular structures. Optionally, we can also use granule clusters instead of individual granules for computing and displaying a cluster density map on the surface of the closest biological structures (Fig. 8). For quantitative analysis, users can interactively explore clusters and their associated attributes, e.g., the number of granules, or total volume. The analysis of glycogen is important for tasks **T2** and **T3**, to analyze glycogen in relation to synapses or mitochondria.

7.4 Implementation

Abstractocyte is implemented in C++ and OpenGL 4.3, using Qt for GUI elements. The application runs on a standard Windows PC and requires a recent GPU for rendering. All case studies were performed on a Windows 10 PC with a NVIDIA GeForce GTX 1070 GPU.

Typical mesh sizes in Abstractocyte are around 8 million vertices with 16 million faces, which can be rendered interactively at more than 40 fps. Our splatting technique for showing proximity values on an object’s surface is also performed dynamically at run time, and achieves framerates of around 10 fps. Clustering of glycogen can be done at run time, and typically takes less than a second for 4,000 granules. Computing the force-directed layout updates for switching from 3D to 2D views is done iteratively whenever users move the selected position along a 3D-to-2D edge in the abstraction space and can be done in real-time as well. We run the force-directed layout algorithm as a background thread to make sure that users can still interact with our system while the layout for the 2D views is being computed.

8 CASE STUDIES

To demonstrate the utility and flexibility of our framework, we present three case studies that we have recorded in user sessions with our collaborating domain scientists. All three cases were performed by a neuroscientist who is working in the field of connectomics and has several years of experience with analyzing electron microscopy data of brain tissue. Prior to the case studies, the neuroscientist had a one hour hands-on training session to get familiar with Abstractocyte.

8.1 Data

For our evaluation and during the initial testing of Abstractocyte our collaborators have analyzed six different electron microscopy volumes. All datasets are from mouse brains, specifically from layer one of the somatosensory cortex. Three volumes are from young adult mice (i.e., around 90 days old), while the other three volumes are from old adult mice (i.e., 3 years old). Each dataset consists of a single astrocyte and roughly 1,000 segmented objects (excluding glycogen granules). For our case studies, our collaborators have used a dataset containing 355 axons and dendrites, 388 spines and boutons, 167 synapses, and 181 mitochondria. In addition to these segmented objects, more than 4,000 glycogen granules have been labeled manually using TrakEM [11].

To prepare the data for our visualization tool, we extract high-quality meshes and skeletons of segmented neurites and astrocytes in a pre-processing step. For extracting meshes, we use NeuroMorph [22], and for computing the skeletons we use a GPU-based framework [25] that implements symmetric parallel and persistence thinning [6]. In addition, we compute the centroid of each object and pre-compute correspondences between each 3D mesh and its skeleton. We compute and store the closest point on the skeleton for each vertex in the mesh, so that we can interactively transition and interpolate between the two.

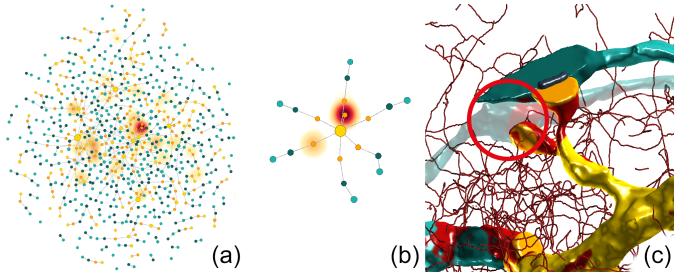


Fig. 9. Case study 1: Evaluating the astrocytic coverage of spines and boutons. (a) Node-link diagram showing neurites as nodes and synapses as links (yellow: dendrites, orange: spines, blue: boutons, cyan: axons). The astrocytic coverage of neurites is shown in the node-link diagram as a heat map; (b) The user selects a node (spine) with high coverage, and filters the graph according to the node’s connectivity; (c) 3D rendering of the spine (orange) that seems to make a synapse with an axon (faded cyan). This synapse was missed in the initial segmentation of our collaborators and only discovered by using Abstractocyte.

8.2 Case Study 1: Astrocytic Coverage of Synapses

Astrocytic coverage refers to how many neuronal structures and substructures are in contact (or lack of contact) with astrocytes. Our collaborators are interested in how many structures are in contact with astrocytes, but also what the percentage of coverage/proximity is for a selected structure, such as a synapse. Astrocytes play an important role in synaptic physiology. For instance, in the recycling of neurotransmitters. This function is crucial for synaptic transmission performance, as excess of neurotransmitter in the synapse might lead to misactivation of neighboring synapses, synaptic over-activation, and eventually cell death. Astrocytic wrapping of the synapse limits these effects, which is why our collaborators wanted to investigate this in their datasets.

The scientist started the session by first getting a high-level overview of the neurites in the dataset using the node-link diagram (Neurite L4, 2D). Next, he looked at the astrocytic coverage of neurites in the node-link diagram, by enabling the heat-map visualization for astrocytes (Astrocyte L4, 2D). He quickly found a spine node with high astrocytic coverage (radiating red node in Fig. 9a). Abstractocyte can quantitatively estimate the total coverage of a structure (in this case the spine) by computing the average distance of all vertices in the spine to the astrocyte. The smaller this value, the closer a structure is to the astrocyte. Next, the scientist examined the spine and filtered the dataset to only contain the spine, its parent (i.e., dendrite) and connected components (Fig. 9b). Then, he switched back to a detailed view of the spine and the connected components (Neurite L0). For displaying the astrocytic coverage, the scientist used the proximity splatting technique in Abstractocyte (Sec. 7.2) (Astrocyte L0). At this point, the scientist was able discover that the selected spine was actually making a synapse with a bouton that had been missed in the labeling process (Fig. 9c), which explained the high astrocytic coverage of the spine. Subsequently, the scientist was able to go back to the original volume data to fix the segmentation of the missed synapse and bouton.

8.3 Case Study 2: Analyzing Glycogen Distribution

In this second case study, our collaborators investigated how glycogen granules are related to boutons and spines (i.e., the locations on neurites where synapses are located). Glycogen granules in astrocytes are energy reserves and are used particularly during sustained activity. Therefore, our collaborators wanted to look at glycogen granule clusters in relation to their closest synapses. By looking at the detailed glycogen distribution and proximity to synapses, they hope to be able to answer the following questions: Does the glycogen distribution correlate with the location of synapses? Does the size of a glycogen cluster correlate with some synapse attributes (e.g., the strength of a synapse)?

Our collaborator wanted to use the same dataset as in the previous case study, so he was already familiar with the general morphology of the astrocyte and neurites. To analyze the glycogen distribution, he first looked at the detailed toon-shaded astrocyte mesh (Astrocyte L1), and enabled the rendering of individual glycogen granules as small

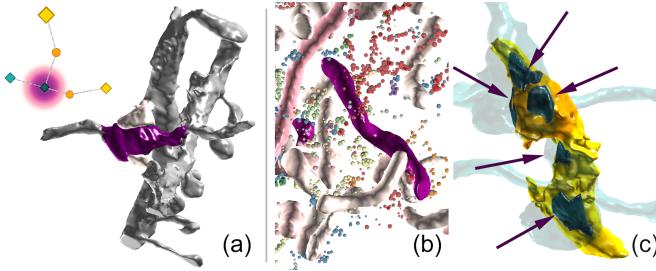


Fig. 10. Case study 2: Glycogen distribution. Glycogen is clustered and mapped to the closest boutons and spines or mitochondria. The node-link diagram (i.e., neurites and synapses) shows the glycogen distribution overlaid as a heatmap; (a) The user selects a bouton with a high value of glycogen mapped to it, and filters the graph based on the connectivity of the selected bouton. The 3D rendering shows that the bouton (magenta) forms two synapses with two different spines (gray), and therefore hypothetically requires more energy; (b) Glycogen mapped to neuronal-mitochondria. The darker, the more glycogen is mapped onto a mitochondria; (c) The same mitochondria belongs to a dendrite (yellow) forming five synapses (black) with different boutons (teal).

spherical objects within the mesh. Next, he first visualized the distance to the glycogen granules on nearby spines and boutons (Neurite L0). After getting a first overview of the glycogen distribution, he performed on-the-fly clustering of the granules and mapped the resulting clusters to their closest spines and boutons. In Abstractocyte, we can visualize this by computing the volume of the glycogen clusters mapped to these spines/boutons and color-code the surface or node accordingly. Next, our collaborator sorted the resulting spines and boutons by their mapped glycogen volumes and selected the top result (Fig. 10a). Upon closer inspection of the identified bouton, our collaborator discovered that the bouton makes two synapses with two different spines, respectively (Fig. 10a). Hence, he continued to investigate the astrocytic coverage of the bouton (similar to case study 1), and determined that the astrocyte wraps around both of these synapses for support. This indicates that boutons that make two synapses might require more energy, which our collaborator now wants to investigate in more detail in the future.

In the final step of this case study, our collaborator was interested in how the glycogen distribution is related to neuronal-mitochondria. To investigate this, he performed the same initial steps as above but mapped glycogen clusters to their closest mitochondria (Fig. 10b). Among the mitochondria that had the highest coverage of glycogen, he was able to identify a mitochondria that belonged to a dendrite that made five consecutive synapses on its shaft, indicating that the operation of these synapses also requires a lot of energy in a small spatial area (Fig. 10c).

Prior to Abstractocyte, our collaborators had to resort to the generic Blender 3D rendering software for rendering high-quality meshes of the astrocyte and glycogen granules [10]. They had no way of easily analyzing the distance of glycogen granules to their closest synapses, and could not interactively filter their data or switch between different visual representations. Using Abstractocyte, the scientist was able to quickly identify relevant neurite structures and further analyze them.

8.4 Case Study 3: Astrocytic-Mitochondria Coverage

Mitochondria are the site of the cellular respiration process, and are responsible for providing energy to the cell. In astrocytes, however, mitochondria can also buffer calcium, thereby influencing intracellular calcium signaling, and potentially regulating synapse activity. Therefore, our collaborator wanted to use Abstractocyte to explore the location of astrocytic mitochondria in relation to synapse locations. Mitochondria tend to appear in clusters or microdomains in astrocytes. Therefore, the scientist also wanted to explore the intracellular distribution of mitochondria within astrocytes.

In this case study, the scientist first looked at the astrocyte in a skeleton representation (Astrocyte L2), with the mitochondria mapped onto the surface of the skeleton. This allowed the scientist to quickly see the morphology of the astrocyte with highlighted locations indicating mitochondria. This way, our collaborator could look at the actual

distribution of mitochondria within the astrocyte. His questions were about how much of the astrocyte's body has mitochondria in it, and whether mitochondria are distributed evenly or, for example, skip fine structures. In this specific dataset, our collaborator could not identify any mitochondria in fine branches of the astrocyte. However, there were some regions within the astrocyte where mitochondria were clustered. Therefore, he explored that area in more detail, especially the synapses close to that area. For this, he filtered the data using the proximity filtering feature (Sec. 7.2) to the astrocytic-mitochondria. Next, he switched to a node-link diagram (Neurites L4, 2D) of the filtered data, and was able to discover a spine with four synapses to four different boutons, see Fig. 4 (1). Fig. 4 (left) shows how the scientist was navigating inside our abstraction space panel during this last part of the case study. Thus, our collaborator switched back to the 3D rendering (Astrocyte L3, Neurite L0), and could identify a handful of synapses connected to one spine that was in very close proximity to an astrocytic-mitochondria. It is believed that mitochondria of astrocytes can buffer calcium which, in turn, can be released and used to signal and regulate synapse activity. Our collaborator speculated that this might be the case for the identified synapses, and wants to further investigate this.

9 DISCUSSION

During the case studies we also collected qualitative feedback from our collaborators. Overall, the scientists felt that Abstractocyte offers a level of flexibility to their analysis and exploration tasks that they did not have before. They moved freely between different abstraction levels and also regularly used interpolated abstraction values, e.g., semi-transparent surface meshes with underlying skeletons.

They outlined three main advantages: First, because Abstractocyte allows displaying neurites and astrocytes at the same time, but with different abstraction levels, it enables users to preserve the context of one, while investigating details of the other. For example, keeping a skeleton of the astrocyte while investigating neurites and synapses helps to see the relation of the astrocyte to the current structures of interest. Second, the proximity visualization allowed our collaborators to get an immediate impression of which structures are close to a structure of interest. Together with the proximity filtering feature, this allowed them to investigate those structures right away. Third, our collaborators were very impressed with the immediate visual feedback during their glycogen analysis case study. The immediate visual feedback sped up their analysis, and allowed them to explore several different hypotheses without having to dive into more elaborate statistical computations right away. Within just the first hour of using Abstractocyte, our collaborators were able to discover things they had not known about their data before.

Limitations. The main limitation of Abstractocyte with respect to analyzing large EM volumes of astrocytes is that our collaborators have so far only collected datasets containing a single astrocyte. Therefore, Abstractocyte currently does not support filtering for specific astrocytes, or the visual comparison of multiple astrocytes. However, supporting multiple astrocytes in a single dataset is straightforward to add in the future. Another limitation with respect to analyzing EM brain volumes is that our current system focuses on the visual analysis aspect. While we do provide the most important quantitative measures, e.g., for clusters, proximity, or coverage, our focus is not on replacing a fully-fledged statistical analysis. On the contrary, we see Abstractocyte as the tool for neuroscientists to quickly explore initial hypotheses about their data before they continue to a more detailed statistical analysis.

10 CONCLUSIONS AND FUTURE WORK

Abstractocyte allows neuroscientists to visually explore astrocytes and neurites at different levels of detail. It enables scientists to intuitively choose the visualization that is most appropriate for their current analysis task, by offering an easy-to-use user interface that allows them to explore all possible combinations of abstractions. We think that by giving users the power to easily switch between abstraction levels in a continuous way, we encourage them to explore their data in more detail. By using smooth transitions between different abstraction levels we can preserve the visual context and correlations throughout the transitions, enabling users to stay focused on their current task, while still providing

the necessary contextual information. In addition to dynamic visual abstractions, we have incorporated methods for the visual analysis of glycogen granules and mitochondria that allow a more detailed analysis of astrocytes in relation to their neighboring neurites.

We believe that the abstraction space introduced in this paper can be generalized and applied to different visualization problems in a wide variety of domains. In the future, we would therefore like to work on allowing other scientists to explore the space of possible visualizations that is suitable for their domain and their specific kind of data.

ACKNOWLEDGMENTS

This work was supported by funding from King Abdullah University of Science and Technology (KAUST) and KAUST award OSR-2015-CCF-2533-01.

REFERENCES

- [1] A. Al-Awami, J. Beyer, D. Haehn, N. Kasthuri, J. Lichtman, H. Pfister, and M. Hadwiger. Neuroblocks - visual tracking of segmentation and proofreading for large connectomics projects. *IEEE Trans. on Visualization and Computer Graphics (Proc. IEEE SciVis 2015)*, 22(1):738–746, 2015.
- [2] A. Al-Awami, J. Beyer, H. Strobel, 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.
- [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] F. A. Azevedo, L. R. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. Ferretti, R. E. Leite, W. J. Filho, R. Lent, and S. Herculano-Houzel. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *The Journal of Comparative Neurology*, 513(5):532–541, 2009.
- [5] C. Basch. Animated transitions across multiple dimensions for volumetric data. Master's thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2011.
- [6] G. Bertrand and M. Couprise. Isthmus based parallel and symmetric 3d thinning algorithms. *Graphical Models*, 80:1–15, 2015.
- [7] 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 Vis. and Computer Graphics (Proc. IEEE Visualization'13)*, 19(12):2868–2877, 2013.
- [8] U.-D. Braumann, H. Franke, J. Hengstler, J.-P. Kuska, and M. Weber. Graph-based quantification of astrocytes. In *Bildverarbeitung für die Medizin 2006*, pages 379–383. Springer, 2006.
- [9] S. Bruckner, V. Šoltészová, M. E. Gröller, J. Hladůvka, 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] C. Calì, J. Baghabra, D. J. Boges, G. R. Holst, A. Kreshuk, F. A. Hamprecht, M. Srivivasan, H. Lehväsliaho, and P. J. Magistretti. Three-dimensional immersive virtual reality for studying cellular compartments in 3d models from em preparations of neural tissues. *Journal of Comparative Neurology*, 524(1):23–38, 2016.
- [11] A. Cardona, S. Saalfeld, J. Schindelin, I. Arganda-Carreras, S. Preibisch, M. Longair, P. Tornancak, V. Hartenstein, and R. J. Douglas. Trakem2 software for neural circuit reconstruction. *PLoS ONE*, 7(6):e38011, 06 2012.
- [12] W.-S. Chung, N. J. Allen, and C. Eroglu. Astrocytes control synapse formation, function, and elimination. *Cold Spring Harbor Perspectives in Biology*, 2015.
- [13] N. Elmqvist, P. Dragicevic, and J. D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Trans. on Visualization and Computer Graphics (Proc. IEEE InfoVis 2008)*, 14(6):1539–1148, 2008.
- [14] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. Second International Conference on Knowledge Discovery and Data Mining '96*, pages 226–231. AAAI Press, 1996.
- [15] A. Frick, A. Ludwig, and H. Mehldau. A fast adaptive layout algorithm for undirected graphs. In *Proc. DIMACS International Workshop on Graph Drawing*, GD '94, pages 388–403, 1995.
- [16] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Exp.*, 21(11):1129–1164, Nov. 1991.
- [17] S. Gerhard, A. Daducci, A. Lemkadem, 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.
- [18] D. Guillaime, C. Viau, and M. J. McGuffin. Hierarchically animated transitions in visualizations of tree structures. In *Proc. International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 514–521, New York, NY, USA, 2012. ACM.
- [19] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE TVCG (Proc. IEEE InfoVis '07)*, 13(6):1240–1247, 2007.
- [20] B. R. R. Helmut Kettenmann. *Neuroglia*. Oxford University Press, New York, 3rd edition, 1995.
- [21] R. Jianu, C. Demiralp, and D. Laidlaw. Exploring brain connectivity with two-dimensional neural maps. *IEEE Trans. on Visualization and Computer Graphics*, 18(6):978–987, 2012.
- [22] A. Jorstad, B. Nigro, C. Calì, M. Wawrzyniak, P. Fua, and G. Knott. Neuromorph: A toolset for the morphometric analysis and visualization of 3d models derived from electron microscopy image stacks. *Neuroinformatics*, 13(1):83–92, 2015.
- [23] J. Kniss, G. Kindlmann, and C. Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Trans. on Visualization and Computer Graphics*, 8(3):270–285, 2002.
- [24] S. G. Kobourov. Force-directed drawing algorithms. 2004.
- [25] Y. Kuang. A high-throughput skeletonization and mesh generation framework for large segmented volumetric datasets. Master's thesis, IACS, Harvard University, 2016.
- [26] 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.
- [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] D. S. Margulies, J. Böttger, A. Watanabe, and K. J. Gorgolewski. Visualizing the human connectome. *NeuroImage*, 80(0):445 – 461, 2013. Mapping the Connectome.
- [29] T. McGraw. Graph-based visualization of neuronal connectivity using matrix block partitioning and edge bundling. In *11th International Symposium on Advances in Visual Computing*, pages 3–13, 2015.
- [30] 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.
- [31] S. Seung. *Connectome: How the Brain's Wiring Makes Us Who We Are*. Houghton Mifflin Harcourt, 2012.
- [32] N. Smit, A. Kraima, D. Jansma, M. deRuiter, E. Eisemann, and A. Vilanova. VarVis: Visualizing Anatomical Variation in Branching Structures. In E. Bertini, N. Elmqvist, and T. Wischgoll, editors, *EuroVis 2016 - Short Papers*. The Eurographics Association, 2016.
- [33] 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.
- [34] 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.
- [35] J. Sorger, P. Mindek, T. Klein, G. Johnson, and I. Viola. Illustrative Transitions in Molecular Visualization via Forward and Inverse Abstraction Transform. In *Eurographics Workshop on Visual Computing for Biology and Medicine (VCBM'16)*. The Eurographics Association, 2016.
- [36] P. Suwannarat, G. Luna, B. Ruttenberg, R. Raviv, G. Lewis, S. K. Fisher, and T. Höllerer. Interactive visualization of retinal astrocyte images. In *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 242–245, March 2011.
- [37] T. Tsandilas, A. Bezerianos, and T. Jacob. Sketchsliders: Sketching widgets for visual exploration on wall displays. In *Proc. 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI'15, pages 3255–3264, 2015.
- [38] X. Yang, L. Shi, M. Daianu, H. Tong, Q. Liu, and P. M. Thompson. Blockwise human brain network visual comparison using nodetrix representation. *IEEE Trans. on Vis. and Comp. Graph.*, 23(1):181–190, 2017.
- [39] M. v. d. Zwan, A. Telea, and T. Isenberg. Continuous Navigation of Nested Abstraction Levels. In *EuroVis - Short Papers*. The Eurographics Association, 2012.