

NeuroMorphoVis: a collaborative framework for analysis and visualization of neuronal morphology skeletons reconstructed from microscopy stacks

Marwan Abdellah, Juan Hernando, Stefan Eilemann, Samuel Lapere, Nicolas Antille, Henry Markram and Felix Schürmann*

Blue Brain Project (BBP), École Polytechnique Fédérale de Lausanne (EPFL), Campus Biotech, Geneva 1202, Switzerland

*To whom correspondence should be addressed.

Abstract

Motivation: From image stacks to computational models, processing digital representations of neuronal morphologies is essential to neuroscientific research. Workflows involve various techniques and tools, leading in certain cases to convoluted and fragmented pipelines. The existence of an integrated, extensible and free framework for processing, analysis and visualization of those morphologies is a challenge that is still largely unfulfilled.

Results: We present NeuroMorphoVis, an interactive, extensible and cross-platform framework for building, visualizing and analyzing digital reconstructions of neuronal morphology skeletons extracted from microscopy stacks. Our framework is capable of detecting and repairing tracing artifacts, allowing the generation of high fidelity surface meshes and high resolution volumetric models for simulation and *in silico* imaging studies. The applicability of NeuroMorphoVis is demonstrated with two case studies. The first simulates the construction of three-dimensional profiles of neuronal somata and the other highlights how the framework is leveraged to create volumetric models of neuronal circuits for simulating different types of *in vitro* imaging experiments.

Availability and implementation: The source code and documentation are freely available on <https://github.com/BlueBrain/NeuroMorphoVis> under the GNU public license. The morphological analysis, visualization and surface meshing are implemented as an extensible Python API (Application Programming Interface) based on Blender, and the volume reconstruction and analysis code is written in C++ and parallelized using OpenMP. The framework features are accessible from a user-friendly GUI (Graphical User Interface) and a rich CLI (Command Line Interface).

Contact: felix.schuermann@epfl.ch

Supplementary information: [Supplementary data](#) are available at Bioinformatics online.

1 Introduction

The study of the morphology of neuronal cells, that is, their shape, is fundamental to reveal some of the key structural and functional aspects of the brain. Not surprisingly, the pioneering work of Ramón y Cajal at the dawn of modern neuroscience focused on cataloging and classifying the intricate cell shapes that Golgi preparations make visible under the microscope, an effort that has not stopped since then (Grant, 2007). Cell morphology influences cell's function and the network topology in several ways. Cross sectional and surface areas determine intrinsic electrical properties of the

excitable membrane of neurons (Segev and Rall, 1998). The shapes of dendritic and axonal arbors affect how cells process information at the cell and circuit levels. At the cell level, the firing pattern of cells is influenced by the shape of the dendritic tree (Van Ooyen *et al.*, 2002; Vetter *et al.*, 2001) and the distribution of synapses affects how information is integrated by the cell (Destexhe and Paré, 1999; Häusser *et al.*, 2000; Jarsky *et al.*, 2005). At a smaller scale, how the cell membrane is shaped at dendritic spines of excitatory neurons has a direct effect on the properties of synaptic transmission due to compartmentalization effects (Adrian *et al.*, 2014). At the

circuit level, morphologies by themselves can be used to predict the characterization of synaptic pathways (Reimann *et al.*, 2015). In order to build computational models that would help advancing our understanding of how the brain computes is therefore subject to obtaining high fidelity digital representations of the cell morphology. We address this concern and present a high level framework for visualization, analysis and repair of neuronal morphologies that are reconstructed from microscopy stacks.

1.1 Motivation and relevant work

Neuronal morphologies are digitally reconstructed from image stacks either with manual, semi-automated or fully automated methods. The reconstruction process includes preparing and staining the tissue sample, acquiring the images and finally tracing the neurites. The end result of this process is a hierarchical skeletal representation of the branching structure of the neurite arborization described as a set of connected points with their cross sectional radii. Each neurite in the skeleton is composed of a set of non-bifurcating sections, where each section is represented by a list of segments and each segment is defined by two connected points. The reconstructed morphology might contain some additional information such as soma (or cell body) profiles or markers for spine and bouton locations. For a comprehensive review of the existing tracing techniques, the reader is referred to previous surveys (Meijering, 2010; Parekh and Ascoli, 2013). Despite the progress made thanks to the DIADEM challenge (Brown *et al.*, 2011), many algorithms still require users to provide some global information for accurate reconstructions. In any case, the result of automatic algorithms needs human validation.

Implementations of some algorithms are available in the form of open source or freeware tools such as VAA3D (Peng *et al.*, 2010), FARSIGHT (Luisi *et al.*, 2011), Neuromantic (Myatt *et al.*, 2012) and Simple Neurite Tracer (Longair *et al.*, 2011). Commercial solutions for neuronal tracing also exist, such as the AutoNeuron extension for NeuroLucida (Glaser and Glaser, 1990), Filament Tracer from Imaris and Aivia. These tools usually provide visualization capabilities and interactive methods to edit the skeleton paths and correct errors. It is also quite common to post-process reconstructions obtained from these tools to further fix different types of problems for which the tools do not provide adequate solutions, such as sample preparation artifacts (e.g. tissue shrinkage). Post-processing is also critical when the morphologies used in a study come from a publicly available database (Ascoli *et al.*, 2007), as there are no guarantees about the quality of the data.

Morphometric analysis tries to derive representative features of the full cell or individual structures such as total dendritic length, neurite count, maximum and average branching order of terminal points and spine length distribution, to name a few. Classification of cells based on their morphologies is another common analysis. Some data used in analysis is estimated directly from the image stacks instead of using skeletal representations or segmentation of finer structures. Tools specific for performing morphometric analysis are L-measure (Scorcioni *et al.*, 2008), which provides extensive quantitative analysis, but relies on other tools for visualization purposes; NeurphologyJ (Ho *et al.*, 2011), which provides a tracing algorithm customized to the type of analysis it performs; and NeuroMorpho (Jorstad *et al.*, 2015), which allows basic analysis of areas and volumes of segmented surfaces. Additionally, reconstructions of the cell membrane using surface or volume representations are interesting for performing other morphometric analysis (such as estimating the area and volume of somata and spines) and also for creating models essential for specific types of simulations or *in silico* brain imaging

(Abdellah, 2017). When mesh membrane models are not available, deriving a mesh representation of the soma and neurites directly from the morphological skeleton is desirable as well. Different algorithms have been developed to generate surface and volume representations of the membrane. Some techniques derive surface representations of the full membrane as a single polygonal mesh using a spherical approximation for the soma (Lasserre *et al.*, 2012) or more elaborate methods to deal with the lack of information about the actual shape (Brito *et al.*, 2013). The problem of voxelizing surface models at high resolution and full cell extent has been addressed by a different technique (Abdellah *et al.*, 2017b), as none of the previous algorithms can guarantee the watertightness required for direct voxelization of the meshes. A similar challenge to DIADEM has been proposed for the segmentation of ultrastructures from electron-microscopy image stacks (Arganda-Carreras *et al.*, 2015).

Multi-compartment neuron models also take morphologies as input. For such models, dendrites are often too detailed compared to the space resolution required by the cable equations, and morphologies are post-processed to collapse multiple consecutive segments into electrically equivalent cylinders. Tools based on image processing libraries (e.g. ImageJ) allow plenty of extensibility, but the graphical interfaces they provide are less user friendly than commercial and more visualization oriented tools. Some open-source frameworks such as Vaa3D allow users to write their custom plug-ins in C++. Plug-ins for commercial scientific visualization frameworks have a lot of potential for integration with additional extensions, at the cost of depending on a proprietary solution. Regardless of their technical merits, an important limitation of commercial tools specialized in neuronal morphology reconstruction and analysis, apart from their price, is the lack of a programming interface for writing user extensions in the form of *plug-ins*.

The existence of so many different, but connected processes highlights the potential utility of a framework that allows tracing, validation, editing, surface and volume reconstruction, analysis and visualization of neuronal morphologies in a integrated and extensible approach to foster collaboration between research groups through code and data sharing. In particular, to the best of our knowledge, no tool integrates morphology visualization, analysis, mesh reconstructions and voxelization. In this paper, we propose a framework that tries to fill this gap. This framework is designed as a Blender (Blender, 2016) add-on that can be loaded into its GUI and can also be called from the command line for batch and distributed processing. Blender is a free and open-source toolset that features several powerful engines for modeling, physics simulation and interactive rendering.

1.2 Contributions

We present *NeuroMorphoVis*: a light, efficient and user friendly framework dedicated to neuroscientific research with the following features:

1. Visualization, analysis and automated repair of digitally reconstructed neuronal morphology skeletons from optical microscopy stacks.
2. Building highly realistic three-dimensional neuronal somata on a physically plausible basis.
3. Creating high fidelity polygonal mesh models of neurons using the repaired morphology skeletons.
4. Creating high resolution volumetric models of neurons that express their optical and spectroscopic characteristics.

2 Architecture

NeuroMorphoVis is composed of five principal modules for (i) data handling, (ii) simulation of three-dimensional somata profiles, (iii) building, repair and analysis of morphological skeletons, (iv) creation of polygonal surface meshes and (v) reconstruction of volumetric models that reflect the geometrical description of the skeletons. A high level overview of the architecture of the framework is illustrated in Figure 1 and a screen shot of its GUI is given in the Supplementary Material (Supplementary Fig. S1).

2.1 Data handling

The framework is capable of loading morphological skeletons stored in the standard file format (.swc) and a custom one (.h5) as well. The users can also load morphologies associated with a certain neuron identifier (or a gid) in a circuit or even an entire cell target that consists of a group of unique neurons. To unify the data processing operations, the loaded skeletons are rearranged in a unifying structure that facilitates accessing the different components of the morphology. We perform a set of preprocessing operations and filters to extract certain information necessary for the skeleton building process and also to repair morphologies containing artifacts.

2.2 Morphology skeleton repair

Due to tracing errors arising during the digitization process, neuronal skeletons are usually reconstructed with different types of artifacts (Conde-Sousa *et al.*, 2017; Myatt *et al.*, 2012; Rodriguez *et al.*, 2003). These artifacts can potentially introduce serious issues and limitations either for simulation (Hernando *et al.*, 2012; Ramaswamy *et al.*, 2015) or meshing (Brito *et al.*, 2013; Lasserre *et al.*, 2012) applications. Our framework is integrated with a module for detecting and repairing various artifacts that have been reported in the literature or observed by our facility members including neuroanatomists, in silico neuroscientists and also visualization experts. The list is by no means exhaustive, however, it is extended whenever we receive new tickets for suspicious artifacts or get acquainted by others published in recent research studies. We managed to find plausible and efficient solutions to repair the majority of these artifacts, for example:

1. When two neurites or more intersect with each other at their first order sections. We determine the neurite with the largest diameter and consider it the principal one emanating from the soma. The other ones are either connected to the principal neurite or directly extended to the soma origin.
2. When there is an intersection between a neurite and the soma, i.e. some samples that belong to this neurite are located within the extent of the soma. In this case, the intersecting samples are either slightly shifted off axis or totally removed from the branch according to their position.
3. When a neurite is disconnected from the soma. This issue happens when the first segment of a neurite is relatively (with respect to the other neurites and other neurons with the same morphological type) located far away from the soma. A reliable solution to this artifact is adding an auxiliary sample between the first segment and the origin of the soma. The location of this sample is set based on the mean radius of the soma and its direction, otherwise the reconstructed soma profile can be totally deformed.
4. When the radius of a sample located on the initial segment of a bifurcating branch is larger than that of the last segment of the parent.

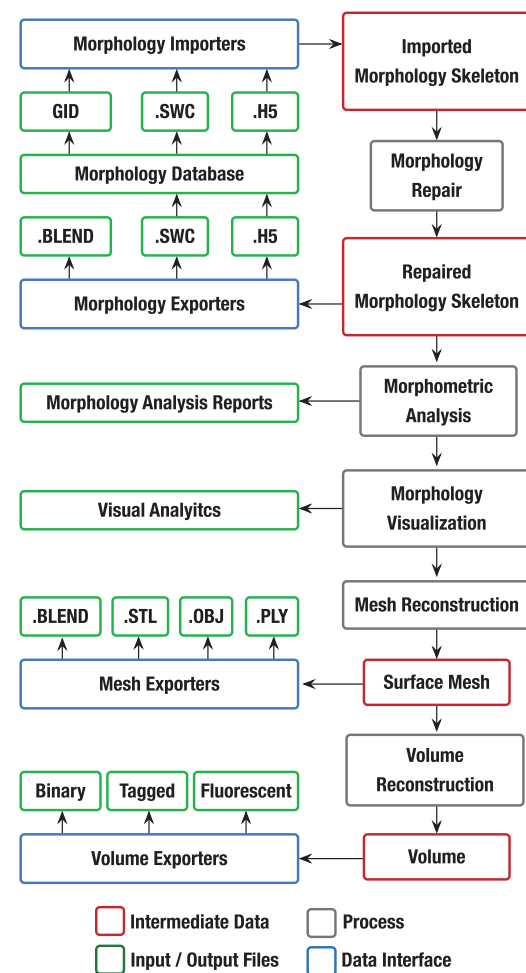


Fig. 1. A high level overview of *NeuroMorphoVis* workflow. The framework imports morphological skeletons from standard file formats and applies a set of predefined filters to repair the skeletons from any tracing artifacts. The user can use the morphometric analysis tools for various purposes, for example, to compute the number of segments per branch or the average segment length per section. The morphology skeleton is converted into a three-dimensional geometric representation based on a specific method selected by the user for visualization and certain analysis purposes. During the interactive visualization session, the user can manually apply other repairs, if required, and perform other visual analytics tasks. Afterwards, the repaired skeleton is used to reconstruct a three-dimensional surface mesh using piecewise watertight meshing (Abdellah *et al.*, 2017b). The resulting mesh can be exported into various file formats for applying simulation data on a vertex basis. This surface mesh is converted into a volumetric model tagged with user-defined optical properties using conservative voxelization. The volume can be exported into common file formats including binary and byte volumes. All the tasks can be executed from a user-friendly graphical interface (GUI) or via command-line interface (CLI) and batch scripts for large scale analysis

A list of all the artifacts the framework could handle are detailed in the documentation. For some other artifacts, which are considered less important, we could not propose convenient solutions to repair them. For instance, when a branch is intersecting with itself. A possible fix to this artifact requires spatial indexing to localize all the self-intersecting branches and then shift them slightly outwards. However, this process might be time consuming for a neuron with complex arborization. All the artifacts detected during the skeleton building process are reported and repaired if requested by the user.

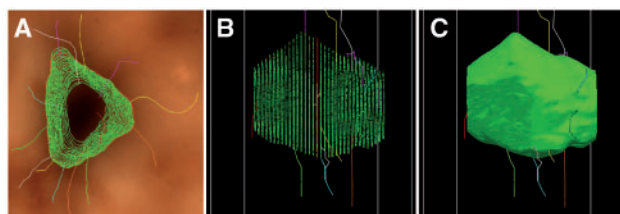


Fig. 2. (A) An overlay of multiple two-dimensional contours of a neocortical pyramidal neuron reconstructed automatically at different depths of fields from a brightfield microscopy stack. (B) A side view of the profiles along the optical axis. (C) The three-dimensional profile of the soma reconstructed from the contours. A smoothing filter is applied to the generated surface to enhance its appearance. The reconstruction and renderings were created with Neurolucida (Glaser and Glaser, 1990)

2.3 Soma profile reconstruction

Due to the fuzzy definition of the soma, the relevant information contained in generic morphology skeletons that describe the soma is usually insufficient to reconstruct a realistic representation of it (Brito *et al.*, 2013; Luengo-Sanchez *et al.*, 2015). In those morphologies, the soma is merely represented by a centroid, a radius that approximates the average distance between this centroid and the initial segments of each neurite, and a projective profile that is traced along a two-dimensional plane (Abdellah *et al.*, 2017b; Halavi *et al.*, 2008; Lasserre *et al.*, 2012). In certain studies, the soma is not modeled based on the reported data in the morphological skeleton, but rather represented by an implicit surface for convenience (Ostroumov, 2007). Therefore, the reconstruction of even an approximation of the soma contour is quite challenging. Recent methods have been presented to provide a univocal definition of the somata, allowing automated characterization of neurons and accurate segmentation of three-dimensional somata profiles measured at multiple depths of fields (Fig. 2) during the tracing procedure (Luengo-Sanchez *et al.*, 2015; Pawelzik *et al.*, 2002). However, this approach can be applied only in advanced reconstructions (Abdellah *et al.*, 2017b).

The soma reconstruction module is added to allow the generation of highly plausible somata profiles relying on their two-dimensional contours and the starting locations of their corresponding neurites. This process simulates the progressive reconstruction of the soma using Hooke's law and mass spring models (Nealen *et al.*, 2006; Terzopoulos *et al.*, 1987). The idea has been adapted from a recent study (Brito *et al.*, 2013) and implemented in Blender (Blender, 2016) using its physics engine (Abdellah *et al.*, 2017b). However, the source code of both implementations was not released publicly. We extended the Blender-based implementation and integrated this module into our framework to provide a convenient tool to validate and compare the somata obtained by segmenting a microscopic stack with the ones extracted from three-dimensional contours. Figure 3 illustrates the process of reconstructing a realistic soma profile from a traditional morphology skeleton, and the procedure is demonstrated in the [Supplementary Video V1](#).

2.4 Morphological skeleton visualization and analysis

Standard skeletons reconstructed with optical microscopes are composed of labeled neurites (axon, basal dendrites and apical dendrite) in addition to the soma, where each neurite is defined by a set of samples that describe their positions and radii in three-dimensional space using a vector-based format (Stepanyants and Chklovskii, 2005). Reconstructed skeletons from electron microscopes are principally stored in volumetric format and segmented later to extract a

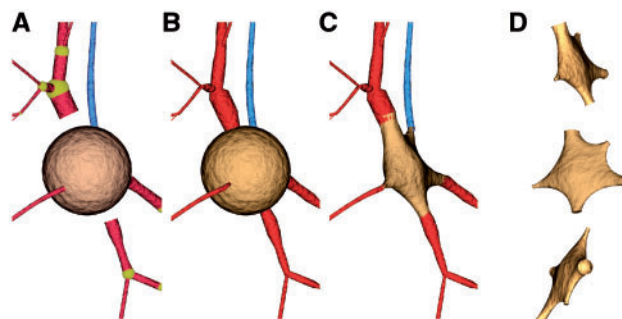


Fig. 3. Accurate reconstruction of a three-dimensional profile of a neocortical neuronal morphology using soft body physics, mass-spring models and a numerical solution of Hooke's law. (A) The soma is initially represented by an ico-sphere. (B) The initial segments of each branch are identified and connected to the origin using an auxiliary sample whose radius is set to zero. On the surface of the sphere, the vertices that correspond to each branch are merged together in a single face and reshaped into a circle with the same radius of the initial sample of the branch. (C) The faces are pulled towards the branches and aligned with the segments that connect their initial segments with the origin. The soft body object is converted into a solid mesh. (D) An orthographic rendering of the front, side and top views of the reconstructed soma

mesh-based representation for each component (Jorstad *et al.*, 2015). Each neurite is structured into a series of sections, each represents a set of connected non-bifurcating segments and each segment is composed of two adjacent samples (Abdellah *et al.*, 2017b).

Visualizing imported skeletons directly from morphology files seems to be a straightforward task if we connect the samples together using poly-lines. However, gleaning insights from these skeletons requires more advanced techniques to highlight certain structural aspects that cannot be revealed from merely connecting the samples. Therefore, we implemented four different algorithms, illustrated in Figure 4, to visualize the various structures of the morphologies either in connected or disconnected styles. The disconnected style is normally used for analysis purposes and the connected one is more suitable for investigating and validating the structure of the entire skeleton following to its reconstruction. It will also be utilized later for creating a polygonal surface mesh that reflects the membrane of the neuron.

2.4.1 Disconnected segments

This style is used to visually analyze the distribution of the segments along a certain section or even an entire branch. Each segment is represented by an independent tapered cylinder whose length is equivalent to the distance between the two samples of the segment. The radius of each side of the cylinder is set to that of the corresponding sample. Two coloring schemes are supported to shade the segments. The first uses alternating colors (black and white) across the whole morphology to visualize the segments distribution, shown in Figure 4A. The default scheme, Figure 4B, applies user-defined colors to each component of the morphology.

2.4.2 Disconnected sections

We also implemented a visualization style that draws a set of connected segments as a single independent section. All the samples that belong to each section in the morphology are converted to a poly-line, whose cross section is defined by a two-dimensional bevel object that approximates a circle. The initial radius of the bevel object is set to one, but the radius of the polyline at each particular sample is multiplied by the radius of the sample itself. With this style,

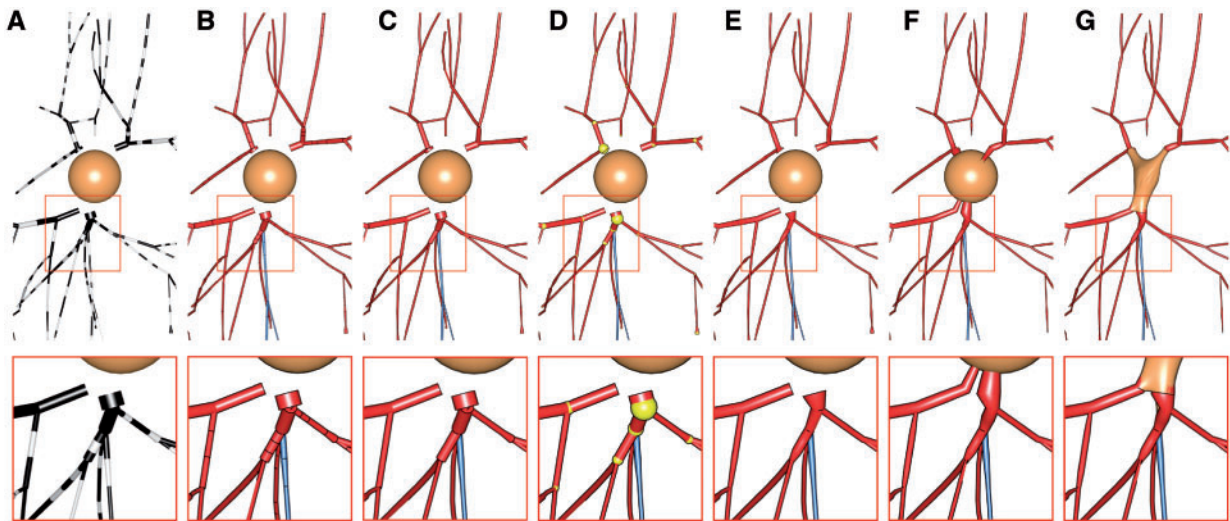


Fig. 4. We implemented different algorithms for visualizing neuronal morphologies. The skeleton is visualized as a set of disconnected segments shaded in alternating colors (black and white) in (A) and using the same colors for each component in (B), where each segment is represented by a tapered cylinder along its two samples. (C) The skeleton is visualized as a set of disconnected sections, where each is drawn as a single polyline. (D) For comparative studies, we implemented a similar method to neuroConstruct where the different sections are joint using uniform spheres (in yellow) (Gleeson et al., 2007). (E) The skeleton is visualized as a set of connected sections, where all the sections that belong to the same branch are drawn as a single polyline object. (F) The skeleton is visualized as a joint object where all the neurites are connected to the soma that is represented by a uniform sphere whose radius is set to the mean of the soma. (G) The soma is reconstructed on a physically plausible basis and connected to all the arbors, and the entire morphological skeleton is integrated into a single object. In (B–G), the soma, basal dendrites and axon are colored in orange, red and light blue, respectively

illustrated in Figure 4C, we analyze the number of segments per specific section selected by the user and the average segment length. Moreover, we added the capability to draw joint spheres between the different sections (Fig. 4D) allowing to compare our skeletons with those created by similar tools, such as neuroConstruct (Gleeson et al., 2007).

2.4.3 Connected sections

Similar to the previous one, we have added another style for drawing the skeleton as a set of connected sections along each branch, as shown in Figure 4E. This style is essential for manual analysis and repair of the branches, where the user can select a specific branch, switch to the edit mode and manipulate the positions or radii of its samples.

2.4.4 Full morphology

In this style, each neurite is drawn as a unique object generated by applying a joint operator on all of its sections. For each neurite, we have added an additional sample at the origin to connect its initial segment to the center of the soma. We used two models of the soma: a symbolic representation by a sphere whose radius is equivalent to the mean radius extracted from the skeleton (Fig. 4F) or by a plausible three-dimensional profile reconstructed as shown earlier (Fig. 4G). This style allows building watertight polygonal mesh models that can represent the membrane of the neuron, and consequently creating high fidelity volumetric models. Other features of this module are demonstrated in the Supplementary Video V2.

2.5 Neuronal mesh reconstruction

To enrich the functionality of our framework beyond visualizing morphological skeletons, we integrated another module to exploit the repaired skeletons for creating high quality polygonal meshes that can accurately represent the surface of neuronal membranes.

This module extends a recent meshing algorithm (Abdellah et al., 2017b) capable of reconstructing piecewise watertight meshes that can be employed to visualize detailed electrophysiological activities obtained from voltage dynamics simulations, where each compartment is mapped to a specific vertex along the mesh (Hernando et al., 2012; Lasserre et al., 2012; Markram et al., 2015).

In summary, a surface mesh representing the whole neuron is created in four basic steps: (i) reconstructing a distinct mesh for the soma, (ii) repairing the morphological skeleton and building a tube representation, as shown in Figure 4G, (iii) meshing each component of the tube model as an individual object, (iv) connecting all the meshes into a single mesh object and optionally (v) decimating this mesh object if desired. The users can adjust the tessellation level of the mesh for optimization purposes. We have used an effective tessellation algorithm that allows the reduction of the number of polygons by 90% and preserving the extent (or the intracellular volume of the neuron). This tessellation level is obtained by comparing the spatial distributions of high resolution volumes created from a reference non-tessellated mesh and other meshes generated at different tessellation levels, shown in Figure 5.

Due to manual tracing artifacts, traditional skeletons are normally reconstructed with hard edges between the different samples. This issue reduces the realism of the generated meshes, in particular for close up renderings. To remove these artifacts, we introduced an option to apply a vertex smoothing filter that smooths those hard edges while preserving the geometry of the rest of the skeleton. Figure 6 shows a comparison between two meshes created with and without smoothing. Finally, the reconstructed meshes can be exported into different standard file formats including the Stanford triangle format (.ply), Wavefront object format (.obj), stereolithography format (.stl) and as a blender file (.blend) for interactive visualization and analysis purposes using other tools in our software ecosystem. Video V3 demonstrates the process of building a high fidelity surface mesh of a neuron from its skeleton.

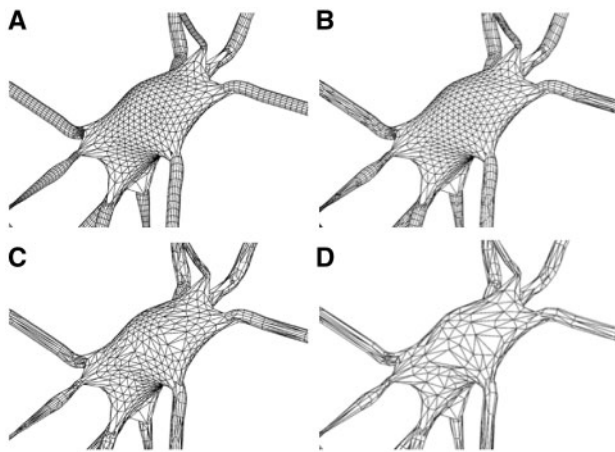


Fig. 5. Wireframe polygonal mesh models of a neocortical neuron created at (A) 100%, (B) 50%, (C) 25% and (D) 10% tessellation levels. Lower level-of-detail meshes are essential for large scale simulation experiments, making it possible to load and visualize a massive amount of neurons on-the-fly and also in volumetric studies where multiple triangles along the surface span the same extent of a single voxel

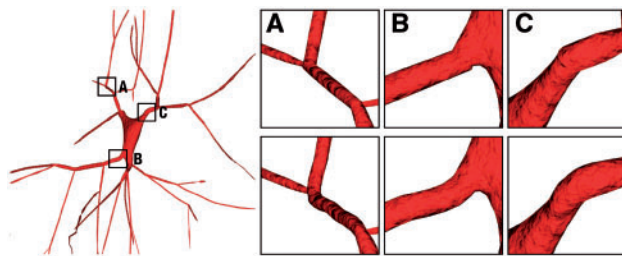


Fig. 6. The effect of using vertex smoothing filters to improve the realism of the reconstructed neuronal mesh models. The shown meshes are generated without (top row) and with (bottom row) smoothing

2.6 Volume reconstruction of neuronal tissue

Volumetric models of neuronal tissue are crucial for performing specific neuroscientific analysis studies (Favre-Bulle *et al.*, 2015; Jarvis *et al.*, 2017; Nikolic *et al.*, 2013) and in silico experiments (Abdellah *et al.*, 2015, 2017a; Markram *et al.*, 2015). They can be also exploited for comparative morphometric studies for validating reconstructed morphologies in volumetric form from microscopy stacks with respect to the ones extracted from vector-based representation. Contrary to geometric ones, those volumetric models can express the intrinsic optical properties of the neuronal tissue, making it possible to accurately simulate light interaction with them using the radiative transfer equation (Pharr and Humphreys, 2010). Creating volumetric representations of neurons from raw morphological skeletons is a challenging task, in particular for large scale neuronal circuits that can contain few hundreds of thousands of neurons or more. A recent solution was presented to accomplish this challenging mission (Abdellah *et al.*, 2017b), allowing the reconstruction of large scale volumetric models of neuronal tissue. This solution was adopted, improved and integrated into our framework as a module that can create volumes of individual components such as spines, neurons (as shown in Fig. 7) or even a high density neuronal circuit containing thousands of neurons (Supplementary Fig. S5). In case of neurons, morphology skeletons are repaired and converted into piecewise watertight meshes, where each section of the morphology is represented by a watertight mesh with zero

non-manifold vertices and edges. The volumes are created from their corresponding surface meshes in two basic steps. The first, called *surface voxelization*, uses conservative rasterization (Hasselgren *et al.*, 2005) to generate volumetric shells that reflect the membrane of the mesh models. The intracellular space of the volumes are filled using *solid voxelization* techniques, for example the recursive eight-way flood-filling algorithm (Burtsev and Kuzmin, 1993). The reconstructed volumes can be stored in binary (one bit per voxel) and byte (8 bits per voxel) formats. The byte volumes are utilized to annotate different structures in the volume either with multiple optical properties, for example for different regions in the brain (Azimipour *et al.*, 2014), or by distinct spectroscopic characteristics to represent different fluorescent dyes expressing the neurons (Swartling *et al.*, 2003).

2.7 Software design and interface

Our framework is written in Python and C++. The data handling, morphology visualization, repair and mesh reconstruction modules in addition to the rendering operations are based on version 2.78c of the Blender API and integrated as an add-on that can be loaded from the GUI. The voxelization code is written in C++ and parallelized using OpenMP, but the functionality is exposed via the user interface to facilitate generating volume models directly with a single click. The framework is designed to run on large scale visualization clusters using distributed computing nodes and multi core CPUs based on Slurm (Yoo *et al.*, 2003). We added support to run the framework from a CLI to make it easy to link it to web interfaces. The tool does not have any specific hardware requirements in general, but the resolution of reconstructed neuronal volumes will be limited by the memory size.

3 Results and discussion

We designed a set of use cases to demonstrate the functionality of each module in the framework using the morphologies of a group of neurons reconstructed from the cortex of a 2-week-old rat. We only report two use cases in this section and present further details on the other ones in the online documentation.

3.1 Visual analysis of somata parametrization

The effect of varying the parameters of the soma reconstruction module, mainly the number of soft body subdivisions and stiffness, is visually demonstrated on a layer V pyramidal neuron. The number of subdivisions was varied between 2 and 6, while the stiffness value was set to 0.001, 0.05, 0.1, 0.5 and 1.0. The number of iterations (or steps) used to run the soft body simulation is fixed to 100 to guarantee convergence. The objective of this case study is to allow the neuroanatomists fine-tuning the simulation parameters until obtaining the most plausible soma profile. A subset of the results are illustrated in Figure 8, whereas the entire matrix that summarizes all the combinations is provided in the Supplementary Material (Supplementary Fig. S2).

3.2 Neocortical volume reconstruction for in silico imaging

Another case study that highlights the significance of *NeuroMorphoVis* in computational neuroscience studies is modeling light interaction with digital models of different brain structures to image the brain *in silico*, i.e. to simulate tissue imaging with optical microscopes. This simulation requires the generation of highly realistic neuronal tissue models that characterize their intrinsic optical

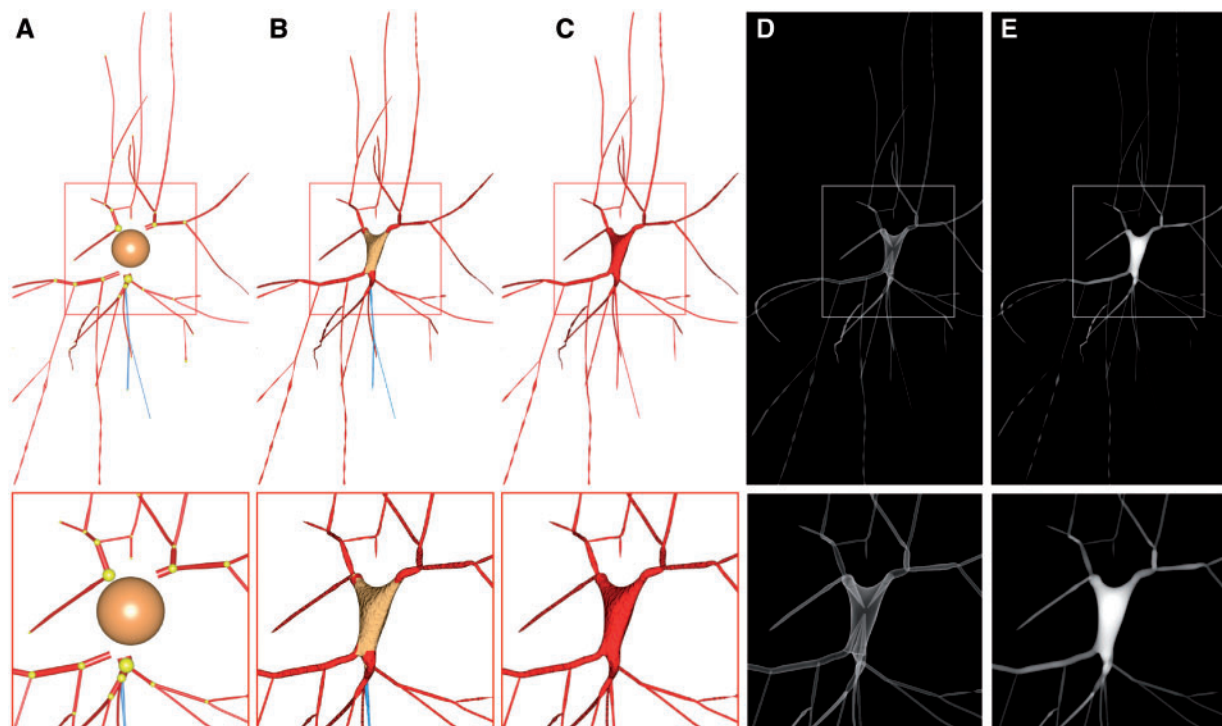


Fig. 7. The sequence followed to reconstruct a solid volumetric model of a neocortical neuron from its morphological skeleton. (A) A geometric representation of the neurites of the morphology is extracted. (B) The three-dimensional profile of the soma is reconstructed on a physically plausible basis. (C) The different components of the neuron (axon, dendrites and soma) are grouped together into a single connected polygonal surface mesh that is piecewise watertight. (D) A volumetric shell of the polygonal mesh is obtained using conservative rasterization and surface voxelization. (E) The intracellular space of the neuron is filled and annotated with user-specified optical properties based on the flood-filling algorithm

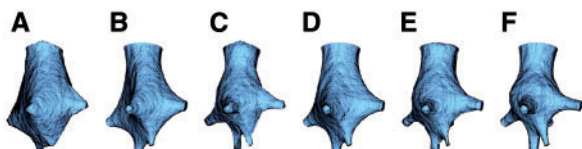


Fig. 8. Three-dimensional mesh models simulating the variations of the soma of a layer V pyramidal neuron using different values for number of soft body subdivisions and stiffness. The simulation parameters are as follows (number of subdivisions, stiffnesses): (A: 2, 0.05), (B: 3, 0.05), (C: 2, 0.5), (D: 4, 0.1), (E: 4, 0.5) and (F: 6, 0.5)

properties, allowing the simulation to be performed on a biophysically plausible basis. This case study demonstrates the functionality of the entire pipeline of our framework starting from loading a group of neurons from a circuit and until the reconstruction of large scale annotated volumes that express the optical properties of brain tissue (Azimipour *et al.*, 2014) and also the intrinsic spectroscopic characteristics of any fluorescent proteins expressed in it. We performed three *in silico* experiments using a single neuron, a group of few pyramidal neurons and a fraction of 10% of a meso-scale circuit that is composed of ~100 000 neurons reflecting a digital slice extracted from the somatosensory cortex of rat brain (Supplementary Fig. S3). The goal of the first experiment is to evaluate and visualize the light distribution within certain tissue region for a specific type of microscopy illumination with varying intensity and wavelength. The result of this experiment is illustrated in Figure 9A. The second experiment simulates the imaging of a group of layer V pyramidal neurons stained with a standard Golgi's method using brightfield microscopy, see Figure 9B. The intracellular space of all the neurons of interest (represented by a specific

target in the circuit) were annotated with the optical properties of Golgi's stain. The last experiment investigates the distribution of neurons tagged with various fluorescent dyes (with different spectroscopic characteristics) that are layer-specific by simulating tissue imaging with widefield fluorescent microscopy. This simulation is used to validate several structural and functional aspects of the circuit, for example, how the fluorescent tags are distributed in the circuit and at which excitation wavelength they can fluoresce or emit fluorescent light. The result of this experiment is shown in Figure 9C. It can also be used to design experimental setups for *in vitro* techniques to refine the input used for data driven modeling.

3.3 Discussion and user feedback

Being an open-source tool written in Python makes our framework accessible to computational neuroscientists, allowing them implementing their algorithms and ideas and share their results in a collaborative way with other teams in our lab or even with external collaborators and interested users. We believe that the add-on design will motivate neuroanatomists with minimal programming knowledge to use the framework in their daily analysis work. The presented use cases have shown an integration of features that do not exist in other frameworks that only focus on morphology analysis or visualization.

To assess the impact and usability of our framework, it was demonstrated to our lab members during an all-hands meeting. This demonstration was followed by an extensive discussion with a group of >10 domain experts including neuroscientists, neuroinformaticians and visualization specialists. We prepared a set of morphologies with various types of artifacts and invited the experts to use our tool to visualize these morphologies and report their feedback in

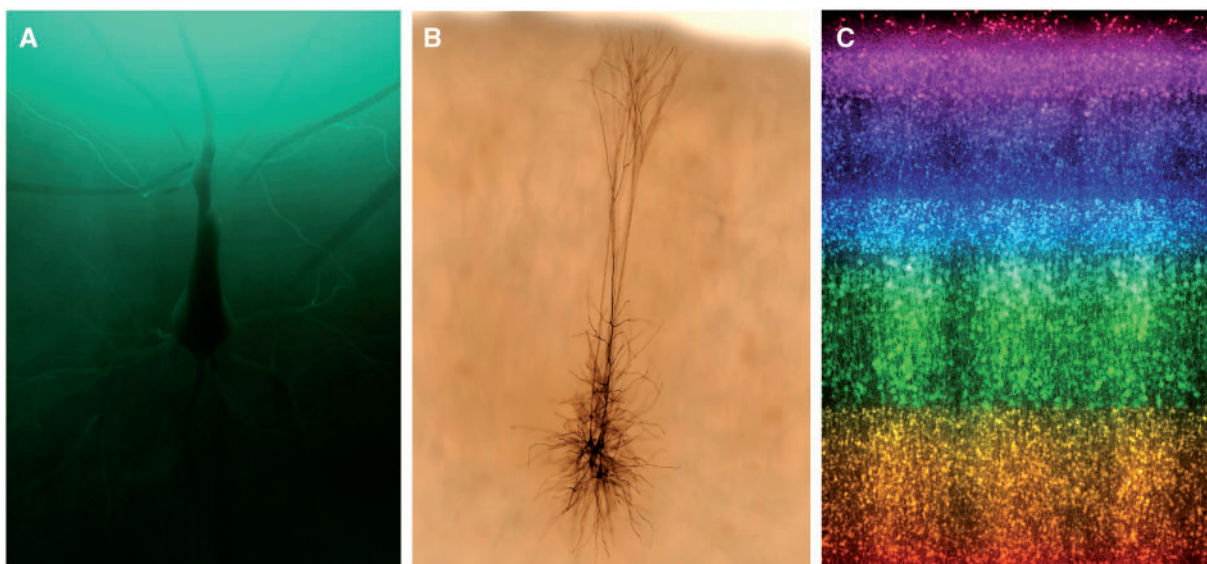


Fig. 9. (A) Volumetric model of a pyramidal neuron immersed in a highly scattering tissue sample. This rendering visualizes the light distribution in the brain tissue at certain light intensity and wavelength. (B) Physically plausible simulation for imaging a group of pyramidal neurons tagged with Golgi's stain using a computational model for brightfield microscopy. The volume is annotated with similar optical properties to Golgi's solution. (C) Simulation of imaging a neocortical slice (~100 000 neurons) labeled with layer specific fluorescent dyes

terms of its functionality, user-friendliness, improvements and feature requests. The interface design was reported to be nice and friendly for domain experts who are familiar with Blender. Experts with limited Blender experience had to reuse the framework few times to get used to it before reporting their feedback. In general, the experts provided a positive feedback. Neuroinformaticians working on morphology synthesis and analysis pointed out that the different methods we have used to visualize the skeletons (shown in Fig. 4) would be extremely helpful to verify the structural and connectivity issues between the different sections along the arbors. The integrated rendering features would allow them creating high resolution visuals that would improve their analysis workflow. The principal idea that was quite exciting for all the domain experts was the ability to visualize, analyze, report broken morphologies and at the end create scientific illustrations and renderings that can be used for their reports and publications in an automated way. The visualization experts indicated that the API provided by the framework can be used to design and integrate novel methods for visualizing and rendering the repaired morphologies and consequently creating highly artistic visuals for their media production. We have received few feature requests including (i) reading morphologies in ASCII file format, (ii) adding a convenient help tooltip for every feature in the user interface to avoid referring to the online documentation repeatedly, (iii) adding support to render the difference between repaired and unrepaired morphologies and (iv) building a web interface, making it possible to load the framework from a web browser.

4 Conclusion and future work

NeuroMorphoVis is an extensible, platform-independent and free framework dedicated to neuroscience research, allowing visual analysis of neuronal morphology skeletons. The framework provides a set of various algorithms and modules for visualizing the different components of a morphology skeleton. It is also capable of repairing skeletal artifacts and constructing high fidelity surface meshes and annotated volumetric models of neurons from their raw skeletons.

The framework is integrated in Blender as an add-on with a user-friendly GUI, providing accessibility to neuroscientists, either experimentalists or theoreticians, with limited programming knowledge. It can be also executed from a rich CLI that accepts configuration files. The functionality of the framework is demonstrated with a set of case studies for constructing three-dimensional soma profiles and simulating the imaging of neocortical neurons with brightfield and fluorescence microscopy.

Software and data availability

The code is available under the GNU General Public License at <https://github.com/BlueBrain/NeuroMorphoVis>. The datasets including morphology files, surface meshes and reconstructed volumes used and created in this paper are available on request. Further information is provided in the Supplementary Material. *NeuroMorphoVis* is designed as a python add-on within the Blender open source software. The add-on depends only on Blender and the HDF5 library.

Funding

Research reported in this publications was funded by competitive research funding from King Abdullah University of Science and Technology (KAUST).

Conflict of Interest: none declared.

References

- Abdellah, M. (2017) In silico brain imaging: physically-plausible methods for visualizing neocortical microcircuitry. Ph.D. thesis, Neuroscience Doctoral School, Ecole Polytechnique Federale de Lausanne, Switzerland.
- Abdellah, M. *et al.* (2015) Physically-based in silico light sheet microscopy for visualizing fluorescent brain models. *BMC Bioinformatics*, **16**, S8.
- Abdellah, M. *et al.* (2017a) Bio-physically plausible visualization of highly scattering fluorescent neocortical models for in silico experimentation. *BMC Bioinformatics*, **18**, 62.
- Abdellah, M. *et al.* (2017b) Reconstruction and visualization of large-scale volumetric models of neocortical circuits for physically-plausible in silico optical studies. *BMC Bioinformatics*, **18**, 402.

- Adrian, M. et al. (2014) Barriers in the brain: resolving dendritic spine morphology and compartmentalization. *Front. Neuroanat.*, 8, 142.
- Arganda-Carreras, I. et al. (2015) Crowdsourcing the creation of image segmentation algorithms for connectomics. *Front. Neuroanat.*, 9, 142.
- Ascoli, G.A. et al. (2007) NeuroMorpho.Org: a central resource for neuronal morphologies. *J. Neurosci.*, 27, 9247–9251.
- Azimipour, M. et al. (2014) Extraction of optical properties and prediction of light distribution in rat brain tissue. *J. Biomed. Optics*, 19, 075001.
- Blender (2016) *Blender: 3D Modelling and Rendering Package*. Blender Institute, Amsterdam.
- Brito, J.P. et al. (2013) Neuronize: a tool for building realistic neuronal cell morphologies. *Front. Neuroanat.*, 7, 15.
- Brown, K.M. et al. (2011) The diadem data sets: representative light microscopy images of neuronal morphology to advance automation of digital reconstructions. *Neuroinformatics*, 9, 143–157.
- Burtsev, S. and Kuzmin, Y.P. (1993) An efficient flood-filling algorithm. *Comput. Graph.*, 17, 549–561.
- Conde-Sousa, E. et al. (2017) N3dfix: an algorithm for automatic removal of swelling artifacts in neuronal reconstructions. *Neuroinformatics*, 15, 5–12.
- Destexhe, A. and Paré, D. (1999) Impact of network activity on the integrative properties of neocortical pyramidal neurons in vivo. *J. Neurophysiol.*, 81, 1531–1547.
- Favre-Bulle, I.A. et al. (2015) Scattering of sculpted light in intact brain tissue, with implications for optogenetics. *Sci. Rep.*, 5, 11501.
- Glaser, J.R. and Glaser, E.M. (1990) Neuron imaging with NeuroLucida—a PC-based system for image combining microscopy. *Comput. Med. Imaging Graph.*, 14, 307–317.
- Gleeson, P. et al. (2007) neuroconstruct: a tool for modeling networks of neurons in 3d space. *Neuron*, 54, 219–235.
- Grant, G. (2007) How the 1906 nobel prize in physiology or medicine was shared between golgi and cajal. *Brain Res. Rev.*, 55, 490–498.
- Halavi, M. et al. (2008) NeuroMorpho.org implementation of digital neuroscience: dense coverage and integration with the NIF. *Neuroinformatics*, 6, 241.
- Hasselgren, J. et al. (2005) Conservative rasterization. *GPU Gems*, 2, 677–690.
- Häusser, M. et al. (2000) Diversity and dynamics of dendritic signaling. *Science*, 290, 739–744.
- Hernando, J. et al. (2012) Towards real-time visualization of detailed neural tissue models: view frustum culling for parallel rendering. In *Biological Data Visualization (BioVis)*, 2012 IEEE Symposium on, IEEE Vis @ BioVis 2012. pp. 25–32.
- Ho, S.-Y. et al. (2011) Neurphologyj: an automatic neuronal morphology quantification method and its application in pharmacological discovery. *BMC Bioinformatics*, 12, 230.
- Jarsky, T. et al. (2005) Conditional dendritic spike propagation following distal synaptic activation of hippocampal ca1 pyramidal neurons. *Nat. Neurosci.*, 8, 1667–1676.
- Jarvis, S. et al. (2017) Neuronal gain modulability is determined by dendritic morphology: a computational optogenetic study. *PLOS Comput. Biol.*, e1006027.
- Jorstad, A. et al. (2015) Neuromorph: a toolset for the morphometric analysis and visualization of 3d models derived from electron microscopy image stacks. *Neuroinformatics*, 13, 83–92.
- Lasserre, S. et al. (2012) A neuron membrane mesh representation for visualization of electrophysiological simulations. *IEEE Trans. Vis. Comput. Graph.*, 18, 214–227.
- Longair, M.H. et al. (2011) Simple neurite tracer: open source software for reconstruction, visualization and analysis of neuronal processes. *Bioinformatics*, 27, 2453–2454.
- Luengo-Sanchez, S. et al. (2015) A univocal definition of the neuronal soma morphology using gaussian mixture models. *Front. Neuroanat.*, 9, 137.
- Luisi, J. et al. (2011) The FARSIGHT trace editor: an open source tool for 3-D inspection and efficient pattern analysis aided editing of automated neuronal reconstructions. *Neuroinformatics*, 9, 305–315.
- Markram, H. et al. (2015) Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163, 456–492.
- Meijering, E. (2010) Neuron tracing in perspective. *Cytometry A*, 77, 693–704.
- Myatt, D.R. et al. (2012) Neuromantic: from semi-manual to semi-automatic reconstruction of neuron morphology. *Front. Neuroinform.*, 6, 4.
- Nealen, A. et al. (2006) Physically based deformable models in computer graphics. In *Computer Graphics Forum*, Vol. 25. Wiley Online Library, pp. 809–836.
- Nikolic, K. et al. (2013) Computational models of optogenetic tools for controlling neural circuits with light. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*. IEEE, pp. 5934–5937.
- Ostroumov, K. (2007) A new stochastic tridimensional model of neonatal rat spinal motoneuron for investigating compartmentalization of neuronal conductances and their influence on firing. *J. Neurosci. Methods*, 163, 362–372.
- Parekh, R. and Ascoli, G.A. (2013) Neuronal morphology goes digital: a research hub for cellular and system neuroscience. *Neuron*, 77, 1017–1038.
- Pawelzik, H. et al. (2002) Physiological and morphological diversity of immunocytochemically defined parvalbumin- and cholecystokinin-positive interneurons in ca1 of the adult rat hippocampus. *J. Comp. Neurol.*, 443, 346–367.
- Peng, H. et al. (2010) V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets. *Nat. Biotechnol.*, 28, 348–353.
- Pharr, M. and Humphreys, G. (2010) *Physically Based Rendering, Second Edition: From Theory to Implementation*, 2nd edn. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Ramaswamy, S. et al. (2015) The neocortical microcircuit collaboration portal: a resource for rat somatosensory cortex. *Front. Neural Circuits*, 9, 44.
- Reimann, M.W. et al. (2015) An algorithm to predict the connectome of neural microcircuits. *Front. Comput. Neurosci.*, 9, 120.
- Rodriguez, A. et al. (2003) Automated reconstruction of three-dimensional neuronal morphology from laser scanning microscopy images. *Methods*, 30, 94–105.
- Scorcioni, R. et al. (2008) L-measure: a web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies. *Nat. Protoc.*, 3, 866.
- Segev, I. and Rall, W. (1998) Excitable dendrites and spines: earlier theoretical insights elucidate recent direct observations. *Trends Neurosci.*, 21, 453–460.
- Stepanyants, A. and Chklovskii, D.B. (2005) Neurogeometry and potential synaptic connectivity. *Trends Neurosci.*, 28, 387–394.
- Swartling, J. et al. (2003) Accelerated monte carlo models to simulate fluorescence spectra from layered tissues. *J. Optical Soc. Am. A*, 20, 714–727.
- Terzopoulos, D. et al. (1987) Elastically deformable models. In *ACM Siggraph Computer Graphics*, Vol. 21. ACM, pp. 205–214.
- Van Ooyen, A. et al. (2002) The effect of dendritic topology on firing patterns in model neurons. *Netw. Comput. Neural Syst.*, 13, 311–325.
- Vetter, P. et al. (2001) Propagation of action potentials in dendrites depends on dendritic morphology. *J. Neurophysiol.*, 85, 926–937.
- Yoo, A.B. et al. (2003) Slurm: simple linux utility for resource management. In *Workshop on Job Scheduling Strategies for Parallel Processing*. Springer, pp. 44–60.