EG UK Computer Graphics & Visual Computing (2022)
M. Turner and P. Vangorp (Editors)

Meshing of Spiny Neuronal Morphologies using Union Operators

Marwan Abdellah[†], Juan José García Cantero, Alessandro Foni, Nadir Román Guerrero, Elvis Boci, and Felix Schürmann[†]

Blue Brain Project
École Polytechnique Fédérale de Lausanne (EPFL)
Geneva, Switzerland

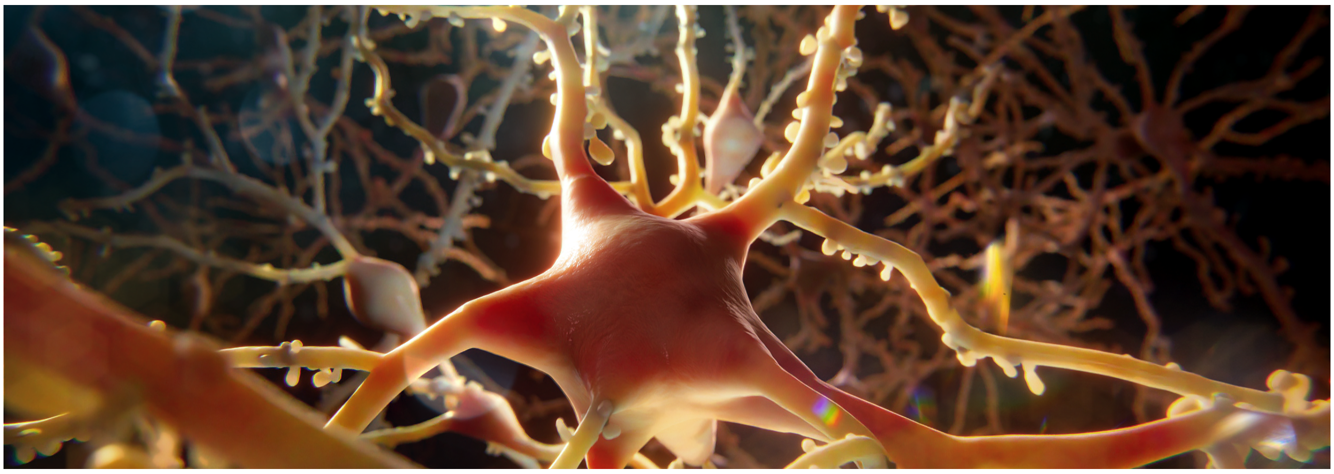

Figure 1: *A network of spiny cortical neurons. The meshes are generated with the proposed algorithm based in the union modifiers in Blender.*

Abstract

Neurons are characterized by thin and long interleaving arborizations in which creating accurate mesh models of their cellular membranes is challenging. While union operators are central for CAD/CAM modeling and computer graphics applications, their applicability to neuronal mesh generation has not been explored. In this work, we present the results of exploring the effectiveness of using union operators to generate high fidelity surface meshes of spiny neurons from their morphological traces. To improve the visual realism of the resulting models, a plausible shape of the cell body is also realized with implicit surfaces (metaballs). The algorithm is implemented in Blender based on its Python API and is integrated into NeuroMorphoVis, a neuroscience-specific framework for visualization and analysis of neuronal morphologies. Our method is applied to a dataset consisting of more than 600 neurons representing 60 morphological types reconstructed from the neocortex of a juvenile rat. The performance of our implementation is quantitatively analyzed, and the results are qualitatively compared to previous implementation. The resulting meshes are applicable in multiple contexts including visualization and analysis of full compartmental simulations and generation of high quality multimedia content for scientific visualization and visual computing (Figure 1).

1. Introduction

Our brain consists of billions of interconnected neurons. Understanding the functional aspects of the brain entails studying the structure-function relationship of individual neurons, in which the structure (or morphology) of a neuron affects the network topology and its dynamics. Visually, neurons are complex structures; they have extremely thin and lengthy branching arborizations that cross different brain regions. They also have multiple morphological (shapes) and electrophysiological classifications that make studying them a computational dilemma, requiring a unifying model to enable full compartmental brain simulations [RCA*15; MMR*15]. Accurate visualization of simulation results depends on the presence of high fidelity neuronal mesh models, with which each mor-

[†] *marwan.abdellah@epfl.ch & felix.schuermann@epfl.ch*

DOI: 10.2312/cgvc.20221168

phological compartment can be mapped to a specific vertex or facet on the mesh surface [LHH*12]. Nevertheless, the generation of those neuronal meshes is a challenging, and in certain cases, computationally intensive task.

We hereby present a simple and effective method capable of creating accurate and detailed mesh models of spiny neurons from their corresponding morphologies using *union operators*. The algorithm is implemented in Blender [Fou16] based on its *Union Modifier*. The implementation is then applied to a diverse set of neocortical neurons that have various morphological types.

1.1. What is a Neuronal Morphology?

Neuronal morphologies are digitally traced and segmented from optical microscopy stacks (brightfield or fluorescence microscopes) of stained tissue slices using semi-automated methods [GSS07]. In other cases, biologically plausible morphologies are synthetically generated in computer simulations based on stochastic sampling techniques and increasingly prescriptive rules inspired from real biological samples [KRS*19]. The end result is a skeleton composed of a series of connected samples. Those skeletons are then stored in a common hierarchical structure representing a directed acyclic graph (DAG). The root node reflects the cell body – or soma – of the neuron. The branching fibers, where the propagation of electrical activity takes place, emanate from the soma. In principle, the morphology is stored in a neuroscience-specific format called SWC as a list of samples. Each sample defines a three-dimensional (3D) Cartesian coordinate and a radius. Each two consecutive samples form a segment, where a series of connected segments between two branching points define a section. Sections are connected to each other in a hierarchical structure, and stored in a tree, where parent and child sections can be accessed from any node in the graph. Figure 2 illustrates the structure of a neuronal morphology.

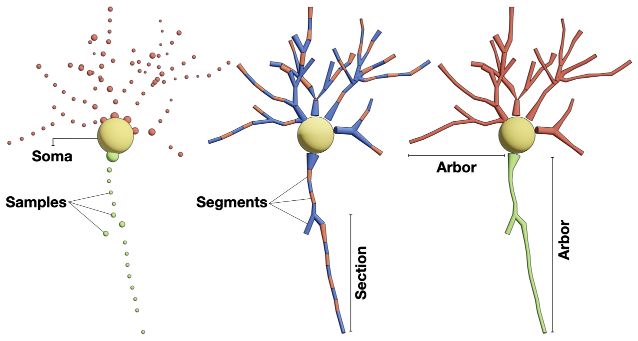

Figure 2: *The structural composition of a neuronal morphology skeleton showing soma, samples, segments, sections and arbors.*

After their reconstruction, neuronal morphologies are processed to remove any artifacts that might arise due to the segmentation process itself. Reconstructed skeletons are then used to perform biophysically detailed compartmental simulations to model their electrical behavior [MCL*06]. Visualizing simulation results cannot be performed on compartmental or simplified geometries, such as spheres and cylinders; the process entails the existence of accurate polygonal mesh models of the neurons that can reflect their membranes [HBB*13]. Nonetheless, creating realistic, smooth and optimized meshes of neurons that can fulfill this objective is not trivial; it is still largely unfulfilled.

1.2. Related Work

Creating mesh models of branching structures, such as trees and blood vessels, is considered an active and important research theme in computer graphics. However, creating polygonal models of neurons from their morphological graphs in particular is comparatively challenging due to their complex structure. Neurons are spatially characterized by extremely large extent with low space occupancy [EBA*12]. They have complex branching structures that cannot be meshed easily and require careful handling and in certain cases pre-processing for successful mesh generation. Reconstructing realistic somatic shapes and having a continuous and smooth connection with the arbors is also a major concern.

Research methods covering neuronal mesh generation can be broadly classified into two categories. The first one is focused on creating optimized, two-manifold and watertight mesh models that can be used to create tetra- or hexahedral meshes for performing reaction diffusion simulations, for example using STEPS [HCWD12] or FEniCS [LMW12] simulators. These methods are required to ensure that resulting meshes are topologically optimized and watertight regardless the realism of their somatic profiles. For example, Mörschel et al. have used a sphere to account for the somatic volume in the simulation [MBQ17]. McDougal et al. [MHL13] have presented a better somatic shape, but the tessellation of their resulting meshes was questionable. Despite being watertight, the meshes belonging to this category do not have natural-looking branching and have high tessellation ($\sim$5–15 million polygons) that would potentially limit their usage in visualization applications to small neuronal circuits with tens or hundreds of neurons on average depending on the application used to visualize them.

The second category is concerned with creating low-tessellated, smooth and realistic meshes that can be used to visualize simulated compartmental activity and render high quality content for scientific dissemination. The principal focus of this category is the tessellation, or polygon count, of the resulting meshes, in order to be able to pack as many neurons as possible in the scene for visualizing large scale neuronal circuits [EBA*12; HBB*13].

Lassere et al. developed an algorithm based on extrusion [LHH*12]. Their implementation builds an initial quadratic proxy mesh around the morphology skeleton. Then, Catmull-Clark subdivision is used to smooth the surface of the proxy mesh. However, their approach was limited in several aspects: (i) the algorithm was implemented in Maya (Autodesk, USA) using its embedded scripting language (MEL) [WK05] which makes it challenging to reproduce the results; (ii) the branching quality was limited; the branches have hard edges and long flaps; (iii) the somatic profile was slightly better than a sphere but still not realistic and (iv) the performance of the technique was poor; it might take more than 15 minutes in some cases to create a single mesh of highly branched pyramidal neuron.

Three methods addressed the reconstruction of realistic somatic profiles from incomplete tracing data. Brito et al. developed Neuronize and presented a novel approach for creating 3D somatic profiles using a physically plausible method based on mass-spring models and Hooke's law [BMB*13]. Their method uses soft body dynamics and pulling forces to transform an initial ico-sphere into an accurate mesh model of the soma. Abdellah et al. have presented a similar approach that uses the physics engine in Blender [AHA*17]. Garcia et al. presented with NeuroTessMesh another approach for building realistic somatic profiles using finite-element modeling [GBM*17]. Their reconstructions were demonstrated on various morphological types to validate the results. Nevertheless, these approaches lacked reconstructing arborizations with realistic and geometrically accurate branching. Creating neuronal meshes with natural-looking branching was presented in a recent study based on skin modifiers [AFH*19]. The results were satisfying for the majority of the cases, but unfortunately, there were some complex branching scenarios (Figure 11b2) that the skin modifier failed to handle.

Contributions Our contributions can be summarized as follows:

1. Based on the recent meshing work of Abdellah et al. [AHA*17], we present a simple method for reconstructing high quality surface meshes of spiny neurons from their morphological graphs using *union operators*.
2. Integrating the implementation in the meshing toolbox of NeuroMorphoVis [AHE*18], to make it freely available to computational neuroscientists, content creators and media specialists.
3. Comparing the results with several previous implementations from [GBM*17], [AFH*19], and [AHA*17].

2. Neuronal Mesh Reconstruction with Union Operators

Boolean operations (intersect, difference and union) are principally used in geometric modeling in two cases: to create sophisticated polygonal models that cannot be modeled otherwise or to design models employing several manual editing [UMC*19]. Our method uses union operators to: (i) bridge secondary branches to primary ones along each arbor in the morphology; (ii) connect each arbor independently to the somatic model; and (iii) attach spine meshes along the surface of aspiny (without spines) neuronal mesh.

2.1. Arbors Reconstruction

Our algorithm creates a single and continuous mesh object per arbor. As detailed previously, neuronal arbors (or branches) are composed of a list of connected sections arranged in a hierarchical manner. Geometric extents of parent and child sections are *partially* overlapping at their terminal samples. Therefore, performing a union operation on each parent-child pair could produce a single, however discontinuous, mesh with intermediate gaps and overlapping branching geometries as illustrated in Figure 3.

Instead of creating one mesh object per section, where the union operation can be performed, we define a list of *primary sections* that represents the most natural continuation from a parent section to a child one along the arbor, and another list of *secondary sections* that branches off the primary ones. This scheme has been implemented before in piecewise-watertight meshing [AHA*17] and extrusion-based meshing [LHH*12]. This labeling helps to

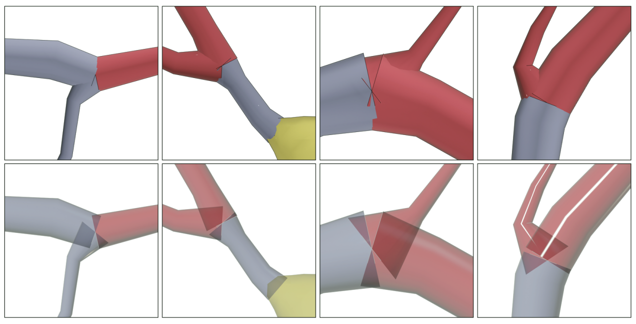
Figure 3: *Overlapping geometries and gaps between the different sections in the morphology. The configurations are rendered with transparency in the bottom row to reveal the overlapping.*

construct independent – and geometrically intersecting – objects, which could be potentially connected using union operators to get rid of the overlapping and intersecting parts to form a continuous and natural looking emanation of child sections from the parent ones at branching points.

Labeling Sections A pre-processing kernel is initially applied to verify if each section in the morphology is primary or secondary. This process can be performed either based on the *angles* between parent and child sections or based on the *radii of their terminal samples* at the respective branching points. It has to be noted that the radii of the terminal samples of the sections that are connected to the same branching point are not necessarily the same. The branch labeling kernel determines the most optimum branching configuration based on a combination of the angle-radius strategies. If the angles between the child and parent sections are comparable, we label the child section with the largest diameter to be primary. This kernel identifies the primary section and adapts the secondary sections accordingly. This adaptation adjusts the radius of the initial sample of secondary sections according to the radius of the last sample of their parent section. This step does not alter the structure of the morphology skeleton per se, but it is essential to avoid any meshing artifacts as shown in Figure 3. Figure 4 illustrates how sections are segregated into primary and secondary to identify principal paths along the arbors.

Resampling Sections In some cases, morphological traces are excessively oversampled in a non-uniform way. This oversampling could be a potential source of error that might introduce geometric artifacts along the surface of the neuronal mesh. To guarantee the generation of a smooth surface (without sharp or jagged edges), another pre-processing kernel is applied to adaptively resample each section in the morphology.

Generating Polylines We recursively process each arbor in the morphology to identify the principal paths starting from root sections that are connected to the soma until reaching the terminal sections, which have no children. Once a path is identified, we construct a corresponding polyline by interpolating a bevel object along the entire path. For example, in Figure 4c, five principal paths are identified and five corresponding polylines are constructed.

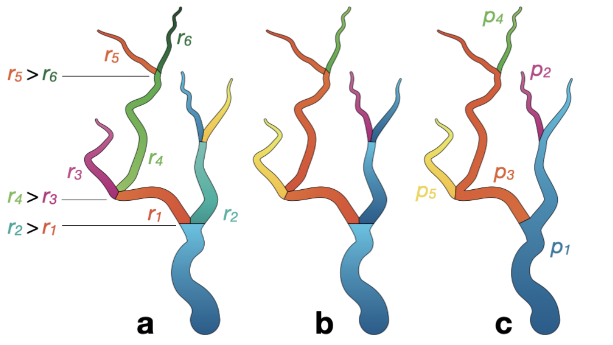

Figure 4: *Labeling sections in the morphology into primary and secondary based on the radii of their terminal samples. The morphology is illustrated (a) before and (b) after the application of the labeling kernel. The principal paths are identified in (c) after the labeling process.*

Generating Arbor Meshes Each polyline is converted into a mesh object and stored in an ordered list. Then, all the mesh objects are welded together one-by-one using the union modifier[†] in Blender to construct the final arbor mesh. Figure 5 shows several closeups on different branches before and after applying union modifiers to bridge the principal paths together.

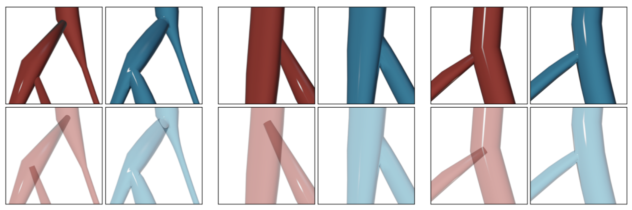

Figure 5: *Applying union modifiers to meshes of principal paths to construct the final arbor mesh. The closeups show the results before (in red) and after (in blue) the application of union modifiers where the overlapping parts are totally removed.*

2.2. Soma Reconstruction

Unfortunately, morphometric analysis of neuronal somata is highly subjective and the definition of the soma is fuzzy [LBB*15]. Moreover, with the technical limitations imposed by optical imaging methods, reconstructing an exact 3D profile of a soma from its imaging data is not always applicable. Somata are typically represented by a centroid and a mean radius approximating the distance between the somatic center the initial morphological sample of each arbor emanating from it. Traces with superior accuracy provide a few more somatic samples called *profile points*, which characterize a two-dimensional projection of the soma along the optical axis of the microscope. This projection might be slightly guiding, but it cannot account for actual 3D profile.

We present an intuitive and efficient algorithm that can reconstruct realistic somatic shapes even if the soma is solely represented by a symbolic sphere. The algorithm is based on implicit surfaces, or *metaballs* [AFZ*21]. It uses a set of auxiliary points from the dendritic sections that are directly connected to the soma to define a 3D profile. Initially, a metaball with a radius equivalent to the mean radius of the soma is created. Starting from the somatic origin, we then build a group of conic meta-segments, each of them connecting the origin to the first sample along the corresponding dendritic tree. The path of each meta-segment is resampled and filled with a series of metaballs with gradually increasing radii. The radii of the first and last metaballs along the segment are set based on the radius of the first sample of the respective dendrite. After the generation of all the meta-segments, all the resulting meta-elements are blended and polygonized to generate a 3D somatic surface that can approximate the natural growing shape of the soma. Figure 6 shows several somatic meshes reconstructed with metaballs.

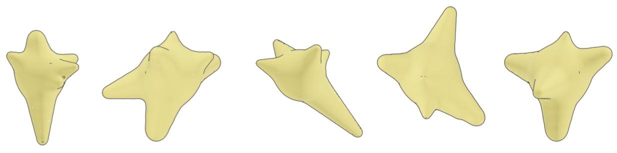

Figure 6: *Generation of somatic meshes of multiple neurons having different morphological types with metaballs.*

2.3. Integrating Spines

Spines are those small protrusions (a few microns size) that emanate from dendritic shafts of neurons to receive an input signal from an axonic terminal of a connecting neuron at a specific site called a synapse. Unlike neurons, spine models are relatively lacking. Spines are segmented into mesh models from volume stacks that are primarily acquired with scanning electron microscopes due to their unprecedented resolutions. Using skeletonization, detailed morphological descriptions of spines are reconstructed from these mesh models. Commonly, morphologies of neuronal reconstructions do not include any spine information. Our circuits [MMR*15] use an algorithmic approach to define spine locations along the dendritic arbors of a neuron. We therefore retrieve the spine information from the circuit and accordingly generate spine mesh models that can be integrated in the mesh reconstructed with our approach. Figure 7 illustrates how spine meshes are integrated along the shafts of dendrites.

2.4. Welding Components

So far, we have created individual and disconnected – but nevertheless overlapping – surface meshes for the soma, dendritic spines and neuronal arbors. We therefore use the union operator to connect, or weld, these components together in order to create a single mesh object that can model the surface of a spiny neuron membrane as shown in Figure 8. This welding process is implemented in two steps. The union operator is first applied on the somatic mesh (as the target mesh) and the polyline mesh of each arbor in the morphology (as the modified mesh) in an iterative scheme. Note that the somatic mesh is only used in the first iteration. The resulting mesh from this iteration represents the merge between the soma and

[†] The union operator in Blender is called union modifier.

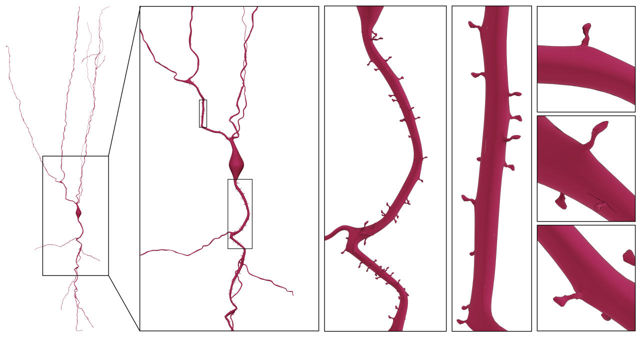

Figure 7: *Spine meshes are grouped into a single mesh with a joint operation. This mesh is then merged with the aspiny neuron mesh with a union operation to create the final spiny neuron mesh.*

the first dendritic arbor. This target mesh will be used in the second iteration as an alternative to the somatic mesh that was used in the first iteration. Therefrom, the resulting mesh from an iteration will be used as the target mesh in the following iteration. After welding all the dendritic meshes to the soma, a single surface mesh of a non-spiny neuron is reconstructed. Depending on the target application, for example: to visualize electrophysiological compartmental simulations, this mesh can be exported without any necessity to integrate the spines along the surface of its dendritic shafts. The integration of the spines can be seamlessly implemented similar to the reconstruction of dendritic arbors in an iterative scheme. However, pyramidal neurons, for example, can have up to tens of thousands of dendritic spines. Consequently, welding the spines one-by-one to the surface of the neuron mesh with such a scheme is computationally expensive and time consuming. The overhead of this process is reduced by grouping the meshes of the dendritic spines into a single mesh object (with multiple partitions) using the joint operator. The union operator is then applied only once on the neuron mesh and the joint mesh of the spines. Geometrically speaking, this process has the same effect of welding the individual spines to the neuronal mesh surface, but it radically improves the performance.

2.5. Smoothing the surface

The surface of the resulting mesh from the welding process is low-tessellated. This mesh is optimum for visualizing scenes with large scale simulations that contain tens or hundreds of thousands of neurons. However, this mesh does not qualify for high quality scientific visualization with closeup shots; there are some sharp and bumpy edges along the surface. We apply a vertex smoothing algorithm to improve the quality of the surface, in particular around the branching regions where the union operations were applied. This algorithm averages the angles between the facets of the resulting mesh. Initially, we clean the mesh by removing any coincidental redundant or duplicate vertices that are relatively located at the same coordinates. Then, we triangulate the mesh by converting all the n-gons (where $n > 3$) into triangles using the triangulation modifier. Afterwards, the mesh is subdivided to increase the density of triangles across the surface. Finally, we smooth the surface by flattening the angles of all the vertices of the mesh in multiple iterations. This process results in a smoothed surface, yet with a high number of

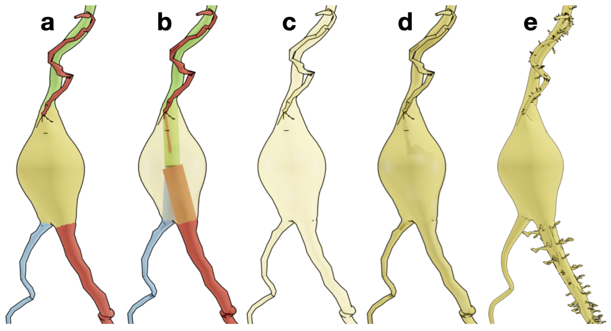

Figure 8: *Welding components to generate a smooth spiny neuron mesh. Each component in the morphology skeleton (arbors and soma) is processed to generate a corresponding mesh (a). The arborization meshes are overlapping with the somatic mesh (b). The union modifier is applied per mesh to reconstruct a single aspiny neuron mesh with smooth connections between the soma and the arbors (c, d). The spines are attached to the surface of the aspiny mesh with another operation to obtain the final mesh (e).*

triangles. So we apply a decimation operation to reduce the tessellation of the mesh to convenient levels comparable to that before the subdivision operation, with which we can load it interactively into any rendering engine. Figure 9 shows a comparison between the surface of the mesh before and after applying the smoothing kernel.

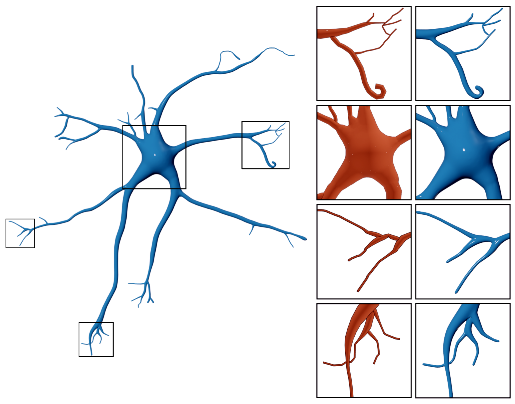

Figure 9: *Smoothing the surface of the reconstructed mesh. The closeups show a comparison of the surface of the neuron mesh before (in red) and after (in blue).*

2.6. Implementation

Our algorithm is implemented in Blender using its Python API. To make it accessible to public, the code has been integrated into NeuroMorphoVis [AHE*18], an open source neuroscientific add-on package for neuronal morphology analysis, visualization and mesh reconstruction. The code can be executed from the graphical user interface (GUI) of Blender or the command line interface (CLI) of the add-on.

3. Results

Our implementation has been assessed with a dataset containing 600 exemplar morphologies that are randomly selected from a recent digitally reconstructed circuit that assembles ∼4,230,000 neurons. The selection reflects 60 different morphological types of neurons that are originally reconstructed from the neocortical microcircuitry of a rodent brain [RCA*15; MMR*15]. Figure 1 shows a rendering of a synthetic neural network where the neuronal meshes are reconstructed with our technique.

3.1. Quantitative Analysis & Error Measurements

The overall precision of the resulting meshes is assessed by comparing the deviation of the resulting surface of the mesh with respect to the extents of the morphological samples of the neuron. We measured the distance between each morphology sample and its nearest neighboring point along the surface of the mesh. The measured values are normalized to evaluate the relative error across the entire morphology. The values are mapped using the heat color-map as shown in Figure 10.

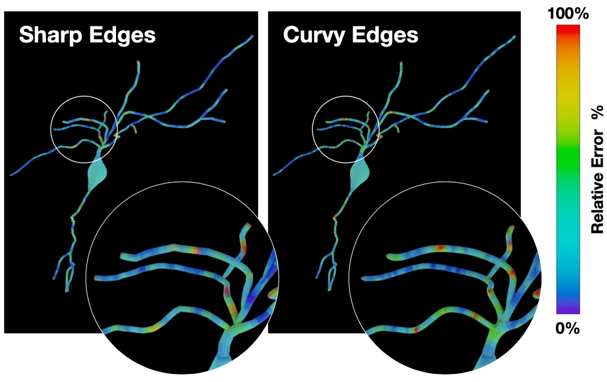

Figure 10: *Error estimation represented by the difference between reported morphology sample diameter and distance from the reconstructed mesh surface expressed as percent error for sharp and smooth meshes created using our algorithm.*

3.2. Comparison with Previous Approaches

Union operators are extremely powerful to yield high quality meshes with smooth surfaces compared to other meshing techniques. This is demonstrated with the results shown in Figure 11 comparing three other meshing techniques that have open source implementations. For example, NeuroTessMesh [GBM*17] is GPU-accelerated, but the branching quality of the resulting meshes is limited even with increased tessellation. Moreover, the soma reconstruction approach seems questionable; the arbors do not emanate smoothly from the soma as they negatively overlap with it. The piecewise meshing technique [AHA*17] produces overlapping geometries that cannot be used to visualize any simulations with transparent colormaps. Skin modifiers [AFH*19] are capable of creating high fidelity meshes with natural looking branching and smooth surfaces, but they fail in several complex branching scenarios and do not provide smooth extrusion of the arbors from the soma. Our method is capable of handling all the limitations of these prior techniques providing: natural-looking branching in all cases, including dendritic spines, smooth surfaces, and continuous soma-to-arbor connections.

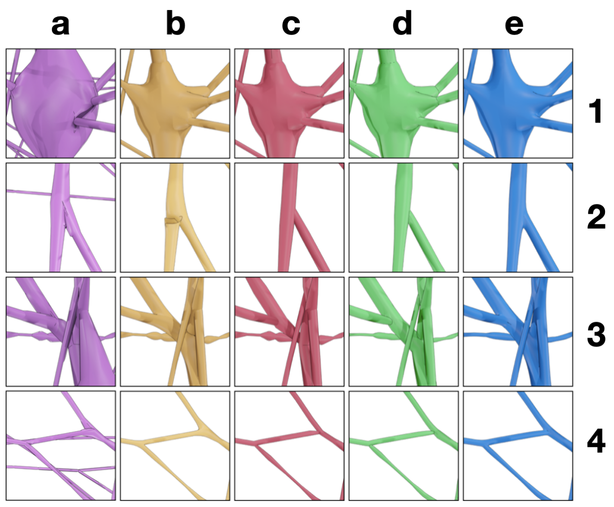

Figure 11: *A comparison between the results of meshing a pyramidal neuron obtained with (a) NeuroTessMesh [GBM*17], (b) skinning modifiers [AFH*19], (c) piecewise watertight meshing [AHA*17], and our implementation with union modifiers without (d) and with (e) surface smoothing.*

3.3. Performance Analysis

The performance of our implementation is assessed with a dataset that contains one neuron per morphological type. Detailed benchmarks and running times of every stage in the algorithm, shown in Figure 12, are generated on a mid-range machine with an Intel Core i9-10900 CPU running at 2.80 GHz and 64 GBytes of DDR4 memory.

Although union operators are powerful in modeling complex branching scenarios, they are known to be relatively slow and compute-expensive compared to skin modifiers [AFH*19], but nonetheless faster that extrusion-based methods [LHH*12]. The performance of union operations depends in principle on the tessellation of operand meshes. But the performance of our algorithm depends on other nested factors as well. Initially, we use multiple union operations per arbor to weld the meshes corresponding to secondary sections to those of primary ones. The performance of this stage does not depend on the number of samples in the morphology per se, but mainly on its number of sections, in addition to the sampling distance used to resample each section in the morphology a priori. The polygonization of each polyline is almost instantaneous, but the tessellation of the corresponding mesh depends on the number of samples in the polyline. Therefore the main performance bottleneck in this stage will be due to the union operations, depending on the number of sections in the morphology. On average, this stage takes several tens of seconds for a neuron with more than 100 morphological sections (e.g. L23_NGC, L5_TPC:A and L5_TPC:B.

The resulting mesh of each arbor is then connected to the somatic mesh with another union operation, per arbor, to generate a single aspiny mesh before the integration of the spines. Somatic meshes have relatively much less tessellation than those of the arbors. Consequently, this bridging operation takes less time than welding the sections along the arbors. On average, the entire neuron reconstruction operation for a smooth neuronal mesh takes from several seconds to tens of seconds.

4. Conclusion

We presented the results of using union operators to synthesize high fidelity meshes of spiny neurons from their morphological graphs. Our method enables the generation of smooth surfaces with plausible somatic profiles, natural-looking branching geometries and decent tessellation, making it possible to build meshes for visualizing electrophysiological compartmental simulations and rendering high quality scientific media content as shown in Figure 1. The quality of our meshes is assessed and compared to the results of other recent methods. The implementation is made freely accessible in NeuroMorphoVis, a widely used tool in the neuroscience community for neuronal visualization and analysis.

Data

Our implementation can be tested by downloading NeuroMorphoVis and selecting the Union method from the Meshing Toolbox. We provide a list of cortical morphologies in the supplementary data. Other cell types are publicly available from the NeuroMorpho.org [ADH07].

Acknowledgments We would like to thank Karin Holm for her valuable comments on the manuscript.

References

[ADH07] Ascoli, Giorgio A, Donohue, Duncan E, and Halavi, Maryam. "NeuroMorpho. Org: a central resource for neuronal morphologies". *Journal of Neuroscience* 27.35 (2007), 9247–9251 7.

[AFH*19] Abdellah, Marwan, Favreau, Cyrille, Hernando, Juan, et al. "Generating High Fidelity Surface Meshes of Neocortical Neurons using Skin Modifiers". *Computer Graphics and Visual Computing (CGVC)*. Ed. by Vidal, Franck P., Tam, Gary K. L., and Roberts, Jonathan C. The Eurographics Association, 2019. isbn: 978-3-03868-096-3. doi: 10.2312/cgvc.20191257 3, 6.

[AFZ*21] Abdellah, Marwan, Foni, Alessandro, Zisis, Eleftherios, et al. "Metaball skinning of synthetic astroglial morphologies into realistic mesh models for in silico simulations and visual analytics". *Bioinformatics* 37.Supplement_1 (2021), i426–i433 4.

[AHA*17] Abdellah, Marwan, Hernando, Juan, Antille, Nicolas, et al. "Reconstruction and visualization of large-scale volumetric models of neocortical circuits for physically-plausible in silico optical studies". *BMC bioinformatics* 18.10 (2017), 402 3, 6.

[AHE*18] Abdellah, Marwan, Hernando, Juan, Eilemann, Stefan, et al. "NeuroMorphoVis: a collaborative framework for analysis and visualization of neuronal morphology skeletons reconstructed from microscopy stacks". *Bioinformatics* 34.13 (2018), i574–i582 3, 5.

[BMB*13] Brito, Juan P, Mata, Susana, Bayona, Sofia, et al. "Neuronize: a tool for building realistic neuronal cell morphologies". *Frontiers in neuroanatomy* 7 (2013). doi: 10.3389/fnana.2013.00015 3.

[EBA*12] Eilemann, Stefan, Bilgili, Ahmet, Abdellah, Marwan, et al. "Parallel rendering on hybrid multi-gpu clusters". *Eurographics Symposium on Parallel Graphics and Visualization*. EPFL-CONF-216016. The Eurographics Association. 2012, 109–117. doi: 10.2312/EGPGV/EGPGV12/109-117 2.

[Fou16] Foundation, Blender. *Blender - 3D modelling and rendering package*. Blender Foundation. Blender Institute, Amsterdam, 2016. url: %7Bhttp://www.blender.org/%7D 2.

[GBM*17] Garcia-Cantero, Juan J, Brito, Juan P, Mata, Susana, et al. "NeurotessMesh: A tool for the Generation and Visualization of Neuron Meshes and Adaptive on-the-Fly Refinement". *Frontiers in neuroinformatics* 11 (2017), 38 3, 6.

[GSS07] Gleeson, Padraig, Steuber, Volker, and Silver, R Angus. "neuroConstruct: a tool for modeling networks of neurons in 3D space". *Neuron* 54.2 (2007), 219–235. doi: 10.1016/j.neuron.2007.03.025 2.

[HBB*13] Hernando, Juan B., Biddiscombe, John, Bohara, Bidur, et al. "Practical Parallel Rendering of Detailed Neuron Simulations". *Eurographics Symposium on Parallel Graphics and Visualization*. Ed. by Marton, Fabio and Moreland, Kenneth. The Eurographics Association, 2013. isbn: 978-3-905674-45-3. doi: 10.2312/EGPGV/EGPGV13/049-056 2.

[HCWD12] Hepburn, Iain, Chen, Weiliang, Wils, Stefan, and De Schutter, Erik. "STEPS: efficient simulation of stochastic reaction–diffusion models in realistic morphologies". *BMC systems biology* 6.1 (2012), 1–19 2.

[KRS*19] Kanari, Lida, Ramaswamy, Srikanth, Shi, Ying, et al. "Objective Morphological Classification of Neocortical Pyramidal Cells". *Cerebral Cortex* 29.4 (2019), 1719–1735 2.

[LBB*15] Luengo-Sanchez, Sergio, Bielza, Concha, Benavides-Piccione, Ruth, et al. "A univocal definition of the neuronal soma morphology using Gaussian mixture models". *Frontiers in neuroanatomy* 9 (2015), 137 4.

[LHH*12] Lasserre, Sébastien, Hernando, Juan, Hill, Sean, et al. "A neuron membrane mesh representation for visualization of electrophysiological simulations". *IEEE Transactions on Visualization and Computer Graphics* 18.2 (2012), 214–227. doi: 10.1109/TVCG.2011.55 2, 3, 6.

[LMW12] Logg, Anders, Mardal, Kent-Andre, and Wells, Garth. *Automated solution of differential equations by the finite element method: The FEniCS book*. Vol. 84. Springer Science & Business Media, 2012 2.

[MBQ17] Mörschel, Konstantin, Breit, Markus, and Queisser, Gillian. "Generating neuron geometries for detailed three-dimensional simulations using anamorph". *Neuroinformatics* 15.3 (2017), 247–269 2.

[MCL*06] Migliore, Michele, Cannia, C, Lytton, William W, et al. "Parallel network simulations with NEURON". *Journal of computational neuroscience* 21.2 (2006), 119 2.

[MHL13] McDougal, Robert A, Hines, Michael L, and Lytton, William W. "Water-tight membranes from neuronal morphology files". *Journal of neuroscience methods* 220.2 (2013), 167–178 2.

[MMR*15] Markram, Henry, Muller, Eilif, Ramaswamy, Srikanth, et al. "Reconstruction and simulation of neocortical microcircuitry". *Cell* 163.2 (2015), 456–492. doi: 10.1016/j.cell.2015.09.029 1, 4, 6, 8.

[RCA*15] Ramaswamy, Srikanth, Courcol, Jean-Denis, Abdellah, Marwan, et al. "The neocortical microcircuit collaboration portal: a resource for rat somatosensory cortex". *Frontiers in neural circuits* 9 (2015). doi: 10.3389/fncir.2015.00044 1, 6.

[UMC*19] Urick, Benjamin, Marussig, Benjamin, Cohen, Elaine, et al. "Watertight Boolean operations: A framework for creating CAD-compatible gap-free editable solid models". *Computer-aided design* 115 (2019), 147–160 3.

[WK05] Wilkins, Mark R and Kazmier, Chris. *MEL Scripting for Maya Animators*. Elsevier, 2005 2.

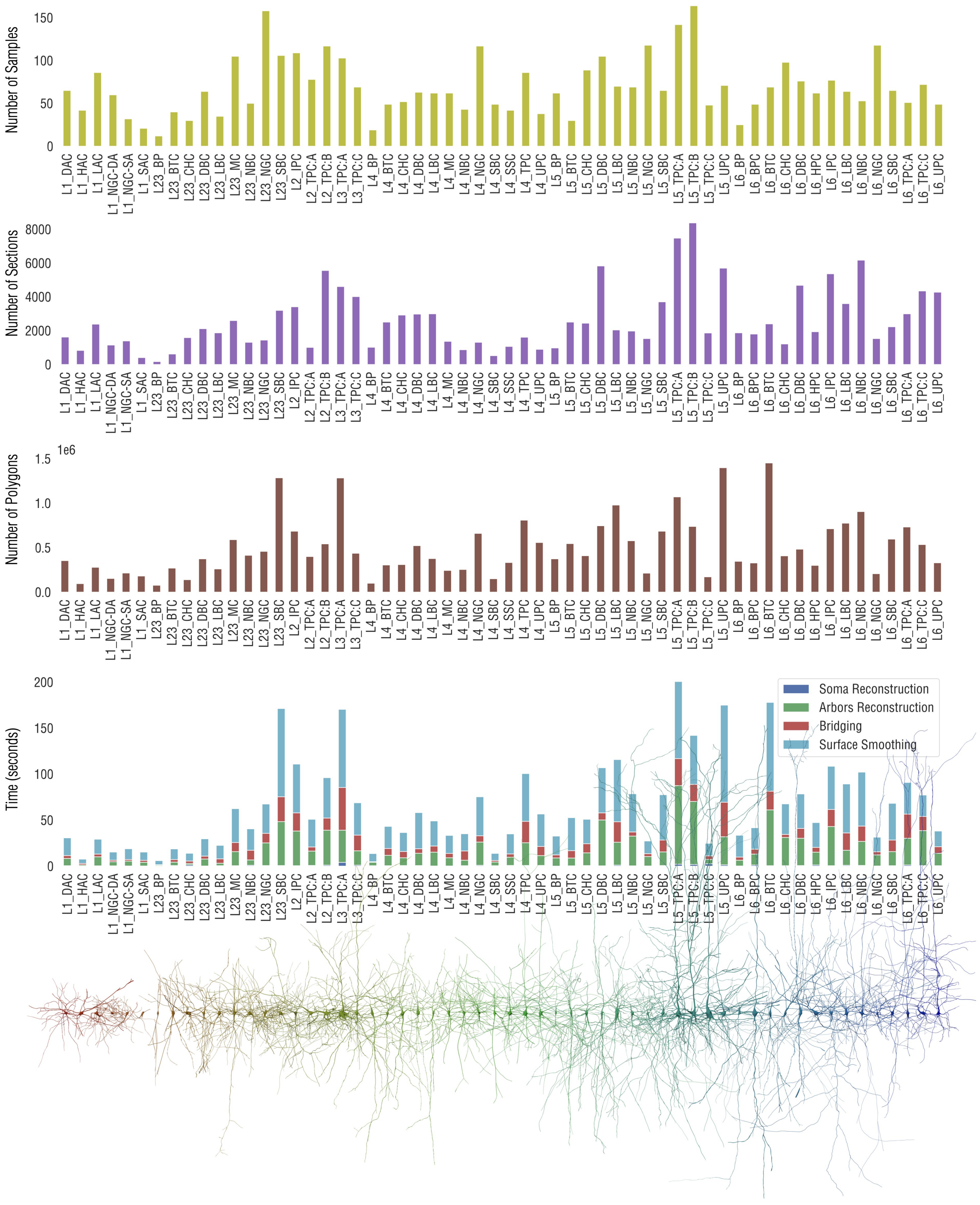

Figure 12: *Performance profiles representing the time (in seconds) of reconstructing surface meshes with our method for a set of exemplar neuronal morphologies reflecting 60 various types of cortical neurons [MMR*15]. The profiles are shown with respect to the number of morphological samples, number of morphological sections and number of polygons in the reconstructed mesh to complete the analysis.*