frontiers

High-quality Neuro Morphology Surface Meshing using Line Skeleton-based Progressive Convolution Approximation

Xiaoqiang Zhu [1], **Xiaomei Liu** [1], **Sihu Liu** [1], **Yalan Shen** [2], **Yimin Wang**[2,*]

[1] *School of Communication and Information Engineering, Shanghai University, Shanghai, China*

[2] *School of Computer Engineering and Science, Shanghai University, Shanghai, China*

Correspondence*:
Yimin Wang
y_wang@shu.edu.cn

ABSTRACT

Creating high-quality membrane surfaces of neural cell morphology for both visualization and numerical simulation is always a challenging task. In this paper, we developed a novel approach of reconstructing water-light 3D neural membrane surfaces from abstract point-and-diameter data. The membrane shapes of the neurons are reconstructed by progressively deforming an initial sphere, and it can be taken as a digital sculpting process. In the dynamic sculpting, the embedded skeleton with radii can serve as a guidance for surface mesh evolution. In order to efficiently deform the surface, a local mapping is adopted to simulate the animation skinning. Therefore, only the vertices within the ROI of the current skeletal position have to be updated. The actual region of influence (ROI) can be determined based on the adopted finite-support convolution kernel, which is convolved with the neural line skeleton to generate a potential field. The progressive convolution potential field guides the mesh evolution to smooth the overall surface regardless of the dendrites or the neural arborizations. On the other hand, the mesh quality during the whole evolution is always guaranteed by the quasi-uniform rules, which splits too long edges, collapses too short ones and moves the vertices within the tangent plane to get regular triangles. Finally, the vertices density on the iso-surface is adaptively distributed according to the neural radii and the surface curvatures.

Keywords: Quasi-uniform mesh, Dynamic sculpting, multiresolution techniques, geometry-based techniques, local mapping query, finite-support convolution kernel

1 INTRODUCTION

With a rapid technical development, increasing attention has been paid to biologically detailed brain visualization and simulation. As the most significant information-processing cells in the brain, neurons are made up of dendrites and axons, which are responsible for receiving and sending signals respectively. Actually each neuron is an electrical entity and various methods have been proposed to simulate the electrical behavior of neurons. However, due to the complexity of neuronal structure, most of these methods simulate the conduction behavior of electrical and biochemical signals in a low-dimensional

space. Recently, there is a growing requirement to simulate cellular behavior in a high-fidelity dimension. Numerical simulation of a reaction-diffusion problem requires a fully-defined neuronal geometry, therefore, a key challenge in neuroscience is to robustly generate a high-quality neuronal membrane surface for chemical-electrical interactions in full neurons or networks of neurons.

From a morphological point of view, the anatomy of neurons can be obtained by interactively tracing neuron elements from microscope images, or accelerated from the microscope image stack using a series of software tools. The neuronal morphology tracking program usually provides a tree-like structure: the unique morphological point in the center of a soma serves as the root node, and the morphology of the neurite is composed of an ordered sequence of interconnected nodes. Besides the 3D coordinates of the skeletal nodes, thickness at each node can also be included to form a point-and-diameter structure. The anatomical features of neurons captured through morphological reconstruction can be used for detailed electrophysiological simulations of voltage dynamics throughout the 3D structure of the neuron (Carnevale and Hines, 2006; Wilson et al., 1988; Gleeson et al., 2007). Unfortunately, the point-and-diameter approximation becomes a problem when one wishes to use the traced structures for problems. Thus, the visualization of such spatiotemporal data poses a particular challenge because of the complexity of neuron morphologies and the limitations of the morphological point-and-diameter representations.

For the neuronal morphologies of tree-like structures, branch modelling is usually the most complicated region of producing neuronal membrane surfaces, as a self-intersection tends to occur at branches. However, we note that smooth branch ramifications were not taken into consideration in most of previous approaches, and the majority of them concentrate on uniform remeshing, which try to generate a whole mesh with the same edge length (Vorsatz et al., 2003; Botsch and Kobbelt, 2004; Kil et al., 2006; Stanculescu et al., 2011). To be able to represent fine geometric details, these approaches inevitably place too many triangles in regions with low-curvatures. To solve these problems, a novel approach is proposed in this paper for robust neural morphology modeling based on skeleton-driven progressive adaptive remeshing, which achieves both smooth surface approximation and high-quality mesh vertex distributions at the branches (cf. Figure 1).

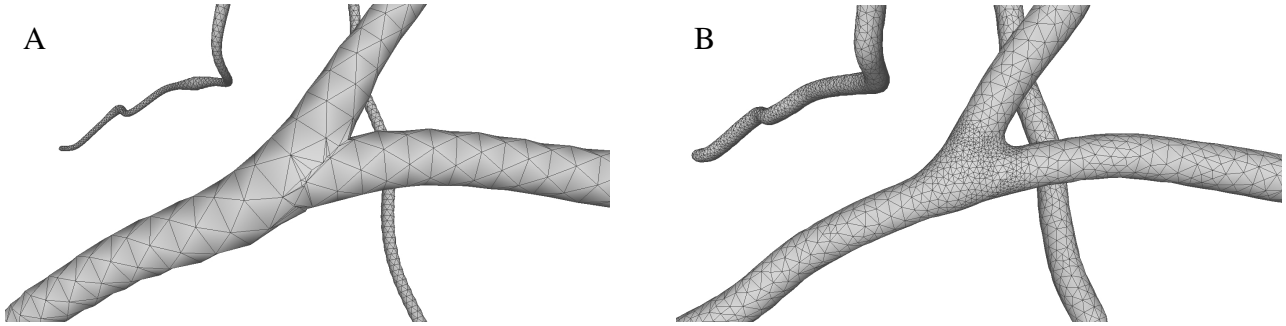

Figure 1. A neuron branch with coarse mesh vertices **(A)** and high-quality adaptive vertex density distributions **(B)**.

This paper presents a sculpting-like mesh generation for 3D visualization of neuron surface and reaction-diffusion simulation applications. A pair of neuronal skeleton and surface mesh are illustrated in Figure 2. In the presented method, the skeletons are firstly loaded from neuronal morphology files, which are then used to drive iterative deformations of the initial triangular mesh. This method allows subsequent on-the-fly adaptive mesh refinements according to different geometric details and it creates high-quality neuronal membranes robustly. Additionally, the water-tight nature of our surface also allows the use of

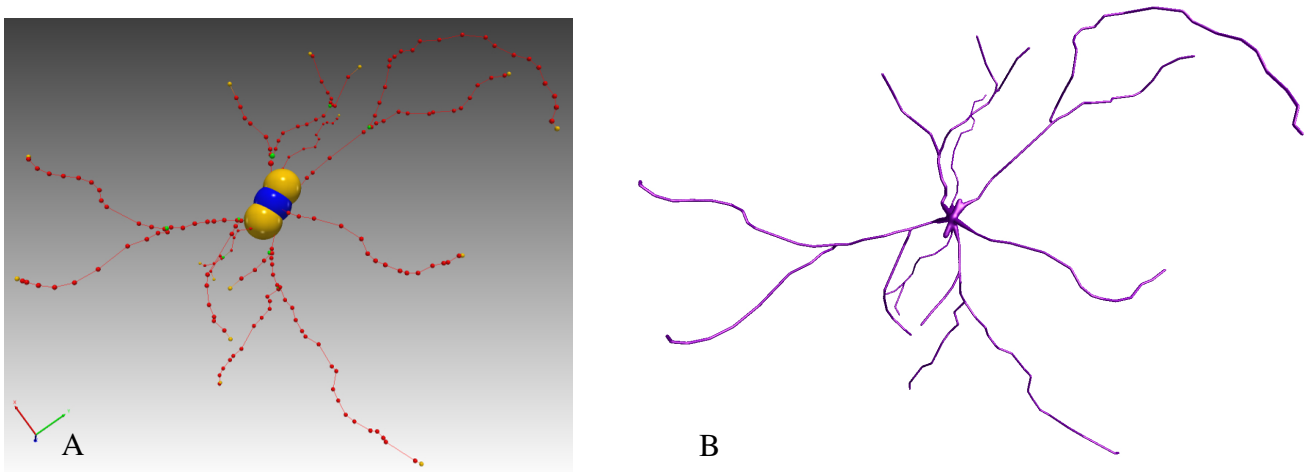

Figure 2. The Morphology of ANDERSON's basal ganglia. **(A)** The structure of ANDERSON's basal ganglia. **(B)** Full mesh.

meshless stochastic diffusion simulators without the risk of ions leaking out of the cell. In summary, the main contributions of this paper are as follows:

1. A high-quality neuron membrane modeling approach is proposed, and the topological robustness can be easily guaranteed through a progressive surface evolution, as the shape of each updated triangle satisfies the adopted quasi-uniform remeshing rules which result in regular triangles.
2. A local convolution surface approximation is presented to modeling branching neuron morphology to achieve a pleasing smoothness and efficiency of the progressive mesh evolution.
3. On-the-fly adaptive mesh refinements are utilized to capture varying scales of geometric details, which balances the mesh quality and the computational expense.
4. A skeletal vertex mapping scheme is introduced to accelerate the neighboring queries, and it also relates mesh vertices to the original morphological skeleton, which is a requirement for other visualization and simulation applications.

2 METHOD

2.1 Architecture

The neuronal membrane mesh is gradually generated through iterative ROI (Region of Influence) queries of an initial model, weighted deformations and local topology reconstructions, which are represented by the half-edge structure in OpenMesh library. The whole workflow consists of the following steps (cf. flow chart in Figure 3).

1. Preconditioning the morphology tree constructed from input files.
2. The meshing algorithm starts along the respective root neurite paths.
3. Querying the ROI in skeletal mappings.
4. Stretching and deforming the mesh in the affected area to the approximated positions.
5. Optimizing the mesh vertex distribution using remeshing with adaptive resolutions according to various geometric details.
6. Iteratively performing the same steps for all neurite paths.

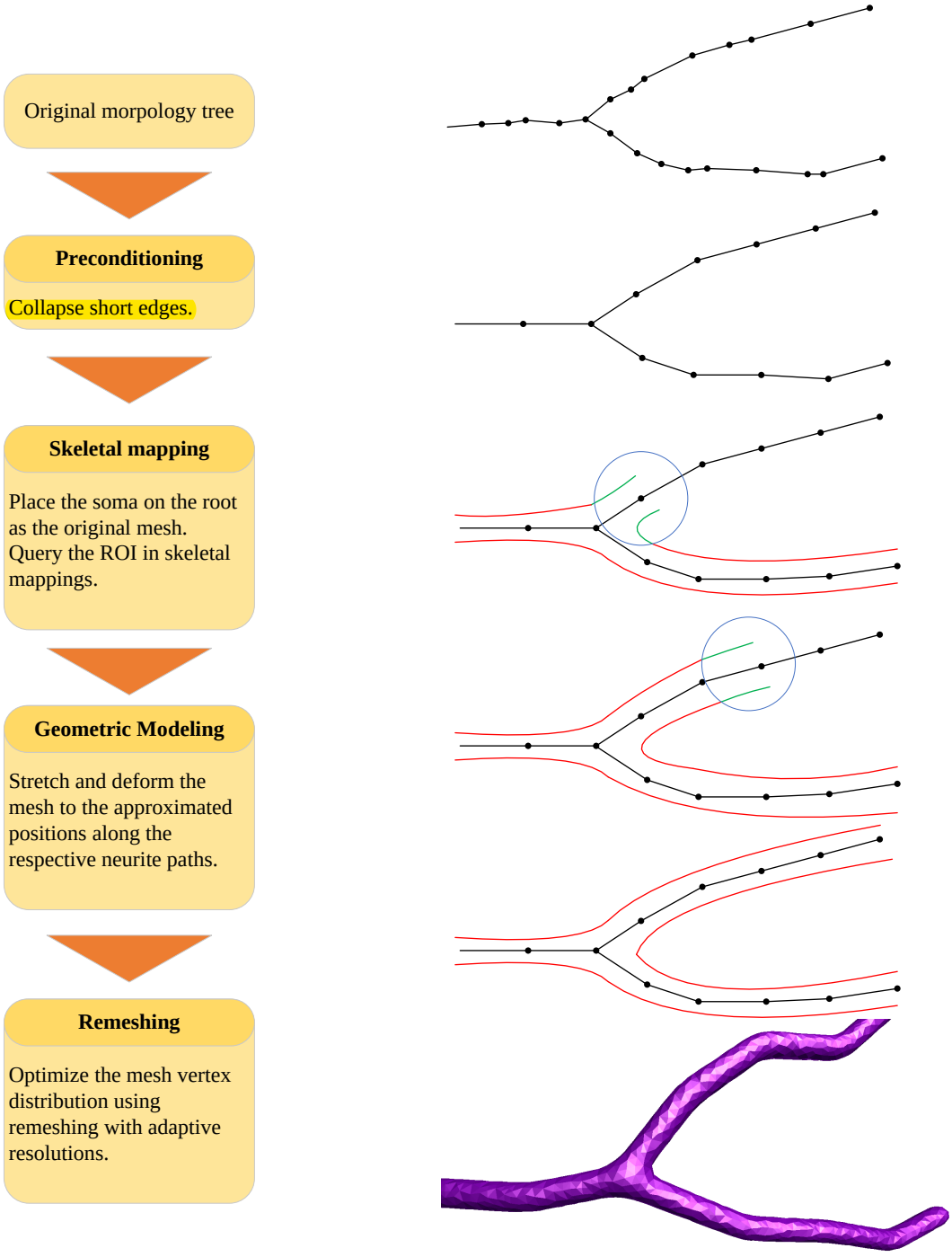

Figure 3. The progressive neuronal morphology modeling workflow. The procedure starts from an initial soma mesh at the root of the neuron tree, then searches the vertices affected by the skeleton-based implicit surfaces, and finally project these vertices onto the implicitly defined convolution surfaces for further remeshing if relevant edges have been longer than the predefined threshold.

2.2 Morphology Preconditioning

Skeletons can be extracted from the central diameters of neurons, and the neurites are usually represented by a series of sequential skeleton segments, with SOMA as the root node. In our architecture, awful cases

occur when the neurite segments are too short compared to the radii of their coincident skeleton vertices. They tend to create irregular meshes which are likely to be self-intersecting and may not be capable of 2D manifold surface generation. However, there are indeed many samples with short neurite segments as well as generally irregular node distribution on NeuroMorpho.Org. Therefore the notion of short neurite segments are to be formalized using violation categories (cf. Figure 3).

2.3 ROI Query Strategy Based on Skeletal Mapping

In order to speed up the generation of neural membrane mesh, a ROI strategy based on skeletal mappings is proposed, which allows local deformation region queries. A list container is attached to each skeleton node for storing those vertex indices which are the closed vertices to the current node. We first query skeleton nodes in the ROI, and then query their mapping vertices on the membrane surface mesh. Compared with the ROI query algorithm based on space partition proposed in (Zhu et al., 2013), our method further accelerates the generation of skeleton-driven mesh modeling. Figure 4 shows the ROI query of skeletal skins.

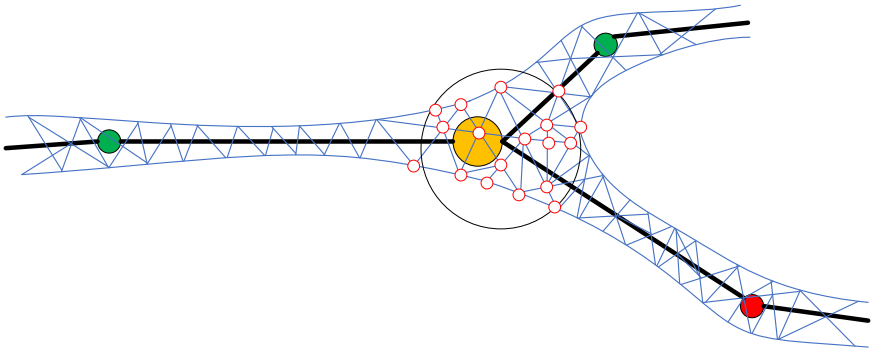

Figure 4. Skeletal mapping.

2.4 Local convolution surface approximation

Skeleton-based implicit surfaces are usually introduced to create branching structures (Bloomenthal and Shoemake, 1991; Hart and Baker, 1996; Jin and Tai, 2002) due to their smoothness and topological variations, however, the Marching cubes polygonizations (Lorensen and Cline, 1987) of the iso-surface they employed suffer from high computation complexity, limited resolution, and low-quality triangular meshes. Furthermore, it is prone to missing small twigs for complex branch models because the output of the Marching cubes is resolution-dependent. Even though there are a large number of improvements (Akkouche and Galin, 2001; Bloomenthal, 1994; Bottino et al., 1996; Van Overveld and Wyvill, 2004; Wyvill et al., 1986; Zhu et al., 2013), it is still difficult to balance the quality of the iso-surface polygons and the performance. Recently, a point skeleton-based metaball policy (Abdellah et al., 2020) is introduced to create accurate morphological models of brain vasculature, and metaballs are also used to skin the different structural components of astrocytes and then blend them in a seamless fashion. However, at straight line skeletons, too many metaballs are essential to placed closely to approximate the cylindrical neuronal or vascular morphologies.

In this paper, our appoach begins with an initial triangular mesh (SOMA), and it will be progressively deformed to approximate the target shape, which is defined as a convolution surface based on embedded

neuronal line skeletons. Therefore, the final neuronal membrane surface mesh achieves high-order smoothness at arbitrary branches.

Convolution surfaces can be regarded as an isosurface embedded in a three-dimensional scalar field, which is calculated by convolving the geometric skeleton with a low-pass filter function. Given a skeleton:

$$g(p) = \begin{cases} 1, & p \in \text{ skeletonV} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

and a kernel function:$f : R^3 \to R$, the contribution of the potential value of the skeleton at point p is the convolution of the function f and g:

$$F(p) = \int_V g(q) f(p-q) dV = (f \otimes g)(p) \tag{2}$$

Among them, the kernel function can be divided into infinite support (such as: Cauchy kernel function (McCormack and Sherstyuk, 1998)) and finite support (such as: quartic polynomial kernel function (Jin et al., 2009)). For a finitely supported kernel function, if the distance of a certain point from the skeleton is greater than the support radius of the kernel function, the potential contribution of the skeleton drops to zero; Instead, for an infinite support kernel function, no matter how far the point in the three-dimensional space is from the skeleton, the potential contribution of the skeleton to the current point is always greater than zero even though it is very tiny. As neuronal morphology usually appears to be a tree-like graph shape, we take line segments as the convolution skeleton, and the fourth-degree polynomial kernel function as the kernel function whose formula can be defined as:

$$f_{\text{Quartic}}(r) = \begin{cases} \left(1 - \frac{r^2}{R^2}\right)^2, & r \le R \\ 0, & r > R \end{cases} \tag{3}$$

where R is the effective radius of the kernel function.

The convolution surface S based on a series of skeleton segments can be defined as:

$$S = \left\{ p \mid \sum_{i=1}^{n} \lambda_i F_i(p) - T = 0 \right\} \tag{4}$$

where F_i is the field contribution of the i^{th} skeletal segment, λ_i is its weight factor, and T is the threshold for extracting the iso-surface from the embedded potential scalar field.

For a point p on the morphological mesh, the adopted local approximation scheme assumes that the closest skeleton to p is a line segment with infinite length. The assumption could not only reduce unessential convolution calculation beyond support radius, but also create natural blending between adjacent line segments due to their smooth thickness variations. Therefore, it is quite suitable for large amount of neuronal morphology approximation with complex graph structures. Actually, the potential contribution of an infinite skeleton to an arbitrary 3D position p can be calculated as:

$$F_{\text{Quartic}}(p) = 2\lambda_i \int_0^{\sqrt{R_i^2 - d_i^2}} \left(1 - \frac{d_i^2 + x^2}{R_i^2}\right) dx = T \tag{5}$$

$$\Rightarrow \lambda_i = \frac{15TR_i^4}{16\left(R_i^2 - d_i^2\right)^{\frac{5}{2}}} \tag{6}$$

where d_i is the average distance between the i^{th} skeleton and p. R_i is the effective radius of the kernel function corresponding to the current skeleton, which can be set to be an empirical value of $2d_i$ in our experiments. And λ_i is the weight of the current skeleton segment, which can be derived analytically.

The solution to the above kernel function is based on the assumption that the value of the kernel function outside a certain distance is infinitesimal. Therefore, the fourth-order polynomial kernel function is usually preferred for its local support characteristics, which could reduce huge convolution computation.

The implementation of approximation is to project each vertex of the remeshed surface onto the implicit convolution surface in their respective gradient directions. The evolution step length of the vertex p can be set as $\frac{1}{2}l_e$, where l_e is the the length of the shortest edge coincident to p. After each subdivision, the projection step length of a new vertex can be derived from its adjacent vertices (Figure 5).

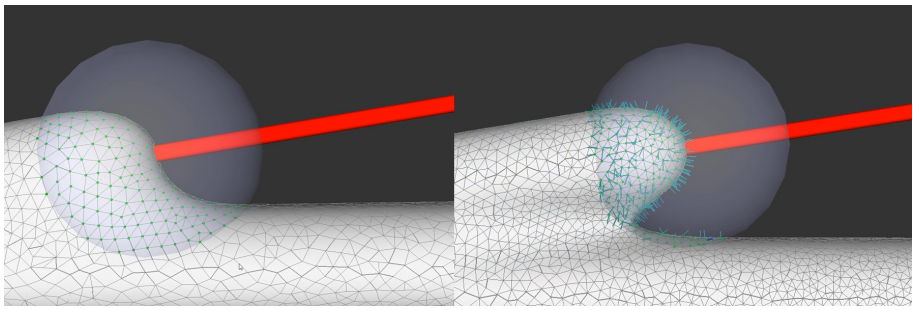

Figure 5. Convolution surface approximation at branches using progressive evolution.

2.5 Remeshing Algorithm based on Quasi-uniform mesh structure

Isotropic remeshing is a valid scheme to optimize the distribution of mesh vertices and topological connectivity, and the surveys in (Botsch et al., 2010; Alliez et al., 2008) have proposed various methods for surface remeshing. Numerous techniques like the particle-based methods (Ahmed et al., 2016), the Delaunay refinement methods (Cheng et al., 2013), and the randomized sampling methods (Ebeida et al., 2016) pay attention to generating isotropic meshes. The methods based on Centroidal Voronoi (Du et al., 1999) tessellation attempt to make each point in accord with the centroid of its Voronoi region. There are various ways to calculate a Voronoi diagram on surfaces, such as mesh parameterizations, discrete clustering, restricted Voronoi diagram (Yan and Wonka, 2015), and geodesic Voronoi diagram (Liu et al., 2016). The fundamental theory of the field-based methods is that isotropic triangular meshes can be taken out from sixway rotational symmetry directional fields (Du et al., 2018). In addition, there are edge-based approaches are based on local operators, including edge split, edge collapse, edge flip, and vertex repositioning (Botsch and Kobbelt, 2004; Dunyach et al., 2013; Wang et al., 2018).

Therefore, in our implicit mesh approximation, two significant aspects will be taken into consideration: 1) A triangular mesh representation of the implicit surfaces should approximate the underlying geometry as accurate as possible; 2) The mesh vertex distribution and connectivity should satisfy high-quality triangles with similar edge lengths coincident to the same polygon, allowing numerically stable calculations.

A quasi-uniform mesh supports successive surface deformation and subdivision at any level of detail, ensuring high-quality triangular mesh during the entire progressive deformation process. Details about how to build and remesh a quasi-uniform mesh will be introduced in the following description. Figure 6 illustrates the dynamic progressive mesh evolution with (right) and without (left) the adopted optimization operations. In the mesh generation, the remeshing based on quasi-uniform configuration can optimize the membrane mesh effectively, thereby achieving robust progressive modeling.

Algorithm Remeshing

Input: $init\_surface, target\_edge\_length$
Output: quasi-uniform mesh
function REMESH($init\_surface, target\_edge\_length$)
  $result \leftarrow init\_surface$
  $low \leftarrow \alpha * target\_edge\_length$
  $high \leftarrow \beta * target\_edge\_length$
  for $vHandle : vHandles$ **do**
    $spilt\_long\_edges(high)$
    $collapse\_short\_edges(low, high)$
    $equalize\_valences()$
    $tangential\_relaxation()$
    $project\_to\_surface()$
  end for
  return $result$
end function

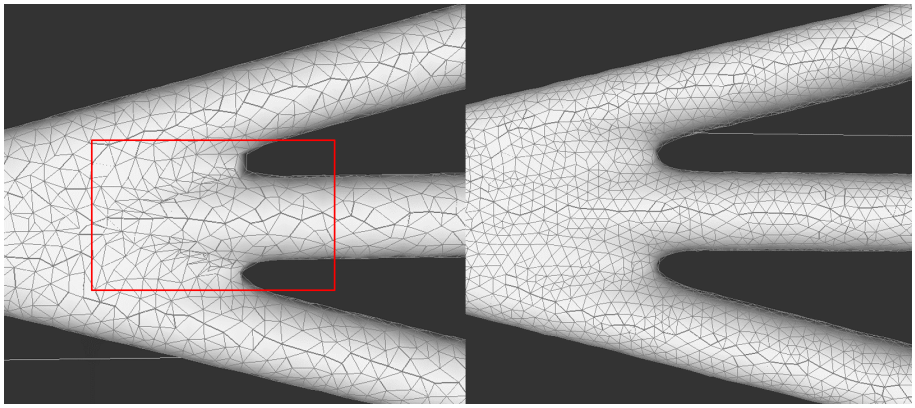

Figure 6. An optimized remeshing based on quasi-uniform mesh.

2.5.1 Quasi-uniform Mesh Configuration

A uniform mesh has roughly the same size elements, which usually appears like Figure 7, and a non-uniform mesh has elements of different sizes. In many aspects, uniform meshes enjoy such advantages as stable structures and efficient high-level geometric algorithms over non-uniform ones. However, creating uniform meshes is non-trivial especially for surfaces with complex topology. Therefore, a relaxation on constraints on uniform meshes is introduced here. A mesh M is said to be quasi-uniform if there exists l_T and d, such that M results from iterative remeshing which assures a maximum edge length l_T and a minimum edge d. That is to say, for each edge $e \in edges_M$, it should falls in the definite interval $length_e \in (d, l_T)$.

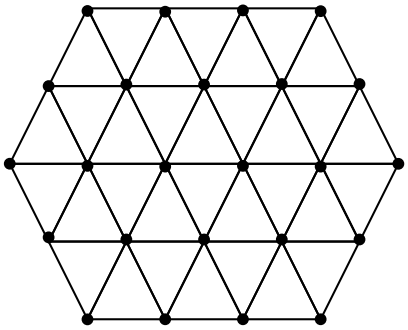

Figure 7. A uniform mesh.

Given a closed manifold mesh M and a threshold l_T , we can ensure that all edges of M are smaller than l_T by iteratively splitting those edges larger than l_T, known as l_T tight mesh. The split operation amounts to adding a vertex at the midpoint of an edge, and correspondingly splitting the relevant triangle face into two adjacent triangles. The iteration over the edges is performed using a simple queue, and the resulting new edges are inserted into the queue.

The l_T tight property is not the only factor to ensure high quality triangles. In most off-line situations, edge collapses can be performed to delete too short edges to generate higher-quality triangle meshes. Edge collapse moves the two vertices of a short edge to the midpoint of the collapsed edge, ensuring that all the edges of M will not be lower than the collapse threshold d (that is, the minimum edge length). Moreover, it is necessary to check the validity of the collapse operation before each implementation. If a collapse operation produces intersected triangles (cf. illegal collapse in Figure 8), the operation is illegal and will not be performed for the current edge.

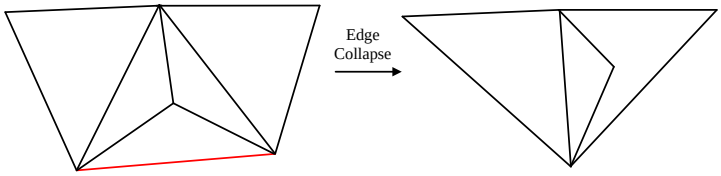

Figure 8. Illegal collapse.

Obviously, it is difficult guarantee that the lengths of all the edges generated after the collapse operation are greater than the minimum length d, and the collapse operation may also destroy the l_T tightness property on the mesh. Therefore, before establishing the l_T tight mesh M, it is necessary to perform the collapse operation to ensure that the minimum length of the mesh edges is greater than d, and then the l_T tightness property should be restored. We note that d can be taken as an internal parameter of a quasi-uniform mesh, since it is bound to $l_T : \mathrm{d} <= l_T/2$, ensuring that the edges greater than l_T will not be divided into edges shorter than d while establishing the l_T tight mesh after a collapse operation.

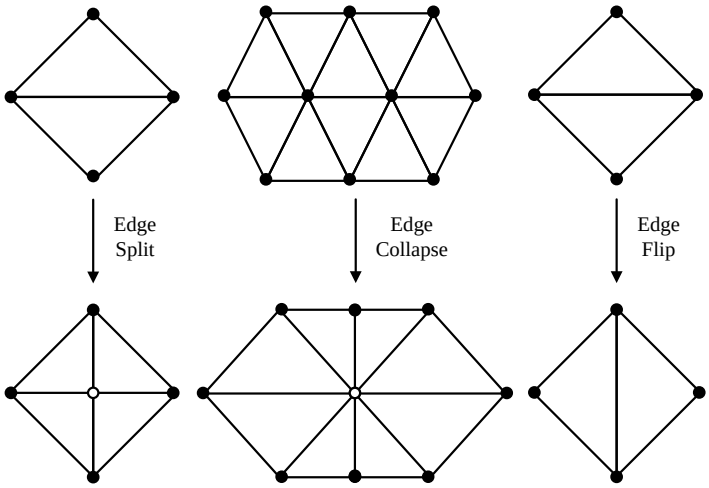

Figure 9. Edge split, collapse and flip.

2.5.2 Area-Equalizing Remeshing

In order to optimize each triangle of a mesh as regularly as possible, an edge flip operation can be applied to the mesh if necessary (cf. Figure 9). The simple measure of the so-called uniform mesh (Figure 7) is: all edges are equal in length; all triangles are equal in area; all internal vertices have a valence of 6, and the boundary vertices have a valence of 4. The valence of a vertex refers to the number of edges attached to the vertex, and it can be balanced through flip operations (Figure 10). An edge flip is performed whenever the operation could approximate the optimal valence. Similar to edge collapses, a legality check is also needed before each flip implementation. Specifically, flips can not be applied to border edges as no enough adjacent triangles are presented. For concave quadrilaterals, reverse faces will be generated after a flip operation as shown in Figure 11, which are also illegal.

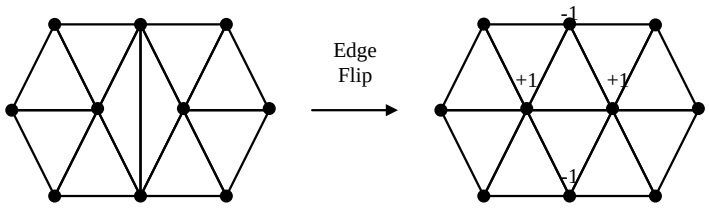

Figure 10. Valence equalization

After iteratively performing the above optimization operations on the mesh M, we can get a relatively uniform triangular mesh. The edge lengths of the mesh are basically close to the target length l, which is between d and l_T, and the valences of the vertices are close to 6. However, the triangle ares, the edge lengths and the vertex valances may still vary across the mesh surface, which can be further optimized. Since k-valence refers to a vertex coincident to k triangles with different areas. Therefore, the averaged area of each vertex contributed by its coincident triangles is dependent of these triangle geometries. To solve this problem, we adopt a fine-tuning by an area-based tangential smoothing (cf. Figure 12). Each

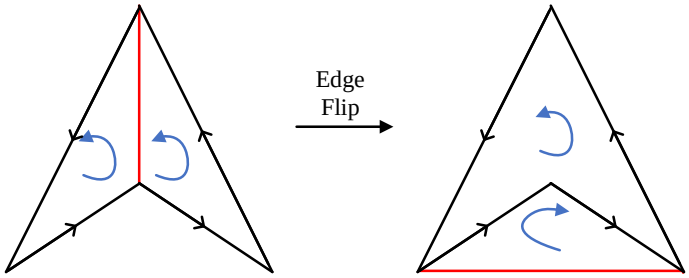

Figure 11. Illegal flip.

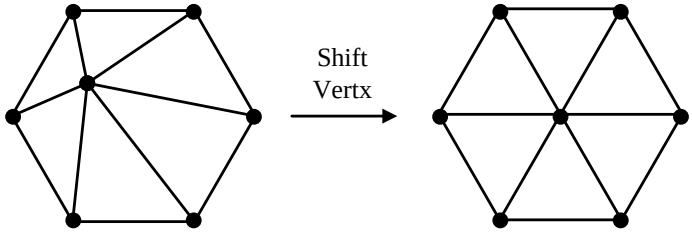

Figure 12. Mesh smoothing.

vertex p_j is assigned a weight that equals to its averaged area $A(p_j)$, and a tangential smoothing moves p_j to its weighted barycentric position

$$g_i = \frac{1}{\sum_{p_j \in N(p_i)} A\left(p_j\right)} \sum_{p_j \in N(p_i)} A\left(p_j\right) p_j. \tag{7}$$

Following each tangential smoothing, a projection onto the implicitly defined surfaces has to be performed for each smoothed vertex (Figure 13), otherwise jitter artifacts maybe occurs. The new position can be updated according to the projection formula

$$p_i \leftarrow p_i + \lambda \left(I - n_i n_i^T\right)\left(g_i - p_i\right), \tag{8}$$

where n_i is the normal vector of p_i and λ is the damping factor for oscillation avoidance. Vertices with relatively large Voronoi regions have more weight, attracting mesh vertices and thus reducing their own area. Experiments show that the mesh area of vertex-related region can be averaged without too many iterations.

2.6 Strategies for adaptive mesh vertex density

To ensure robust generation of high-quality mesh models for neuron morphologies, a mesh optimization for high-resolution levels of details is required. However, a high-resolution representation usually results in too many tiny triangles on the surface parts with low curvatures, which not only imposes high burden on memory but also wastes unnecessary computational time. On the other hand, in order to generate

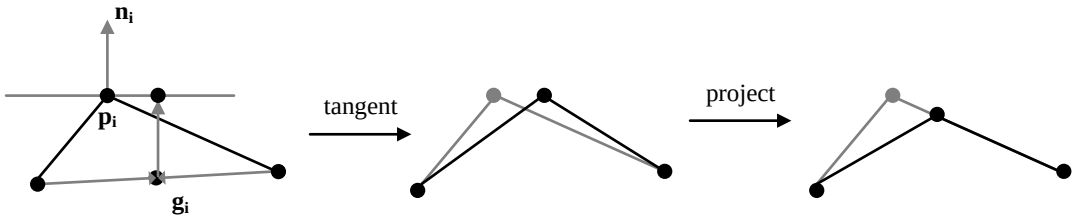

Figure 13. Tangential relaxation and vertex projection.

topologically correct surfaces at branches robustly, a relative high-resolution surface mesh is preferred. Therefore, the marching step size of the mesh evolution should be related to both the skeletal structures and mesh vertex densities so as to create neuron morphology with adaptive edge lengths.

The robustness of the surface mesh generation at branches and the corresponding mesh vertex density depend on the angles between the branches, which are used to control the marching step size along the skeletons. Actually the mesh vertex density can be determined according to the target edge length in the current region, which are indirectly related to the branching angles. It should be noted that the mesh vertices in the ROI ought to be queried as fast as possible for efficient mesh evolution, so the step sizes of both the marching along skeletons and the convolution surface approximation should not exceed the lower length limit of the mesh edges. Details for determining vertex densities at a branch are as follows:

1. **Separating Point.** According to the angle of a neuron branch and the support radius R of the target convolution surfaces, the separating position t at the branch can be calculated. Since the distances between t and the skeleton segments of the both branches are R, the projection points of t on the skeleton segments can be obtained by calculating the tangent value of θ, which are the corresponding positions with the maximum vertex densities in dynamic adaptive mesh evolution (Figure 14).

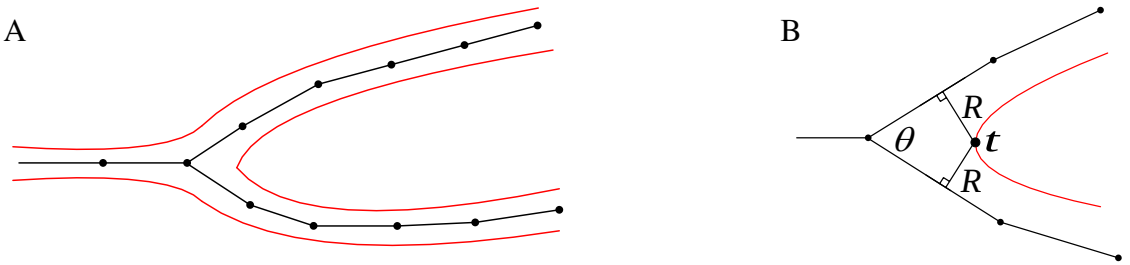

Figure 14. Adaptive mesh resolutions at branches. **(A)** presents a branch, and **(B)** shows the separating position t.

2. **Maximum Vertex Density.** As the neuron morphology at the branching t usually creates a surface with the locally highest curvature, the maximum vertex densities are assigned for the mesh in the nearby region. Therefore, triangles with the minimum edge length are generated in the separating region based on a marching along the skeleton with the minimum step size.
3. **Minimum Vertex Density.** On the other hand, the maximum target edge length can be usually set as the average edge length of the initial mesh, as our mesh evolution begins with an initial soma which

possesses the largest radius. The local maximum edge lengths at other positions that are far away from branching can be calculated according to the ratio of the current skeletal radii and the soma radius.

4. **Edge Length Interpolation.** After the locally minimum target edge lengths and the maximum ones are determined, edge lengths for intermediate places between each pair of the minimum $l_{\min}$ at t and maximum length $l_{\max}$ can be linearly interpolated accordingly.

Moreover, the adaptive mesh evolution could not only optimize the vertex distribution but also accelerate the surface generation and subsequent rendering, as illustrated in Figure 15.

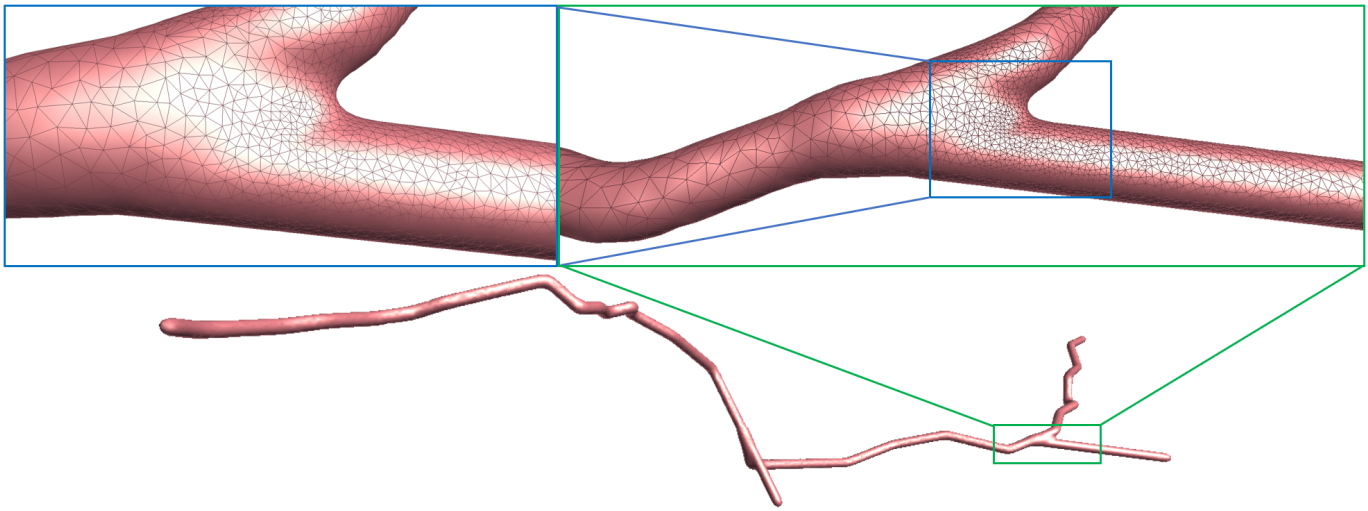

Figure 15. Adaptive mesh vertex distributions in a separating region.

3 RESULTS AND DISCUSSION

3.1 Results

To validate our proposed mesh evolution-based convolution approximation, our method has been applied to many cases from one of the largest cell morphology databases NeuroMorpho.Org, as shown in Figure 16. It is easy to see that the underlying convolution potential field could smooth the whole membrane surface regardless of the dendrites or the neural arborizations. Due to the advantageous superposition property of convolution surfaces, no advanced blending operations are introduced to produce crease-free rounded neuronal surfaces. Moreover, in order to improve the efficiency of the modeling system, we propose the ROI query strategy based on skeleton-vertices mapping, which can effectively accelerate the mesh evolution.

On the other hand, the mesh vertex density distribution is optimized to represent adaptive details (Figure 17). The density of the surface vertices is related to both the thickness of the neuronal morphology and the surface curvature. Although the thickness can be easily obtained from the morphology files, it is nontrivial to calculate the curvature in a dynamic mesh evolution. The main reason lies in that the curvature has to be pre-determined before the quasi-uniform remeshing which is used to the edge length limitations. To this end, our insight is that high curvatures are distributed at branching regions especially at the separating regions of skeleton-based convolution surfaces. Therefore, a proportionate ratio to the distance between an edge and the separating points is employed to the current edge length.

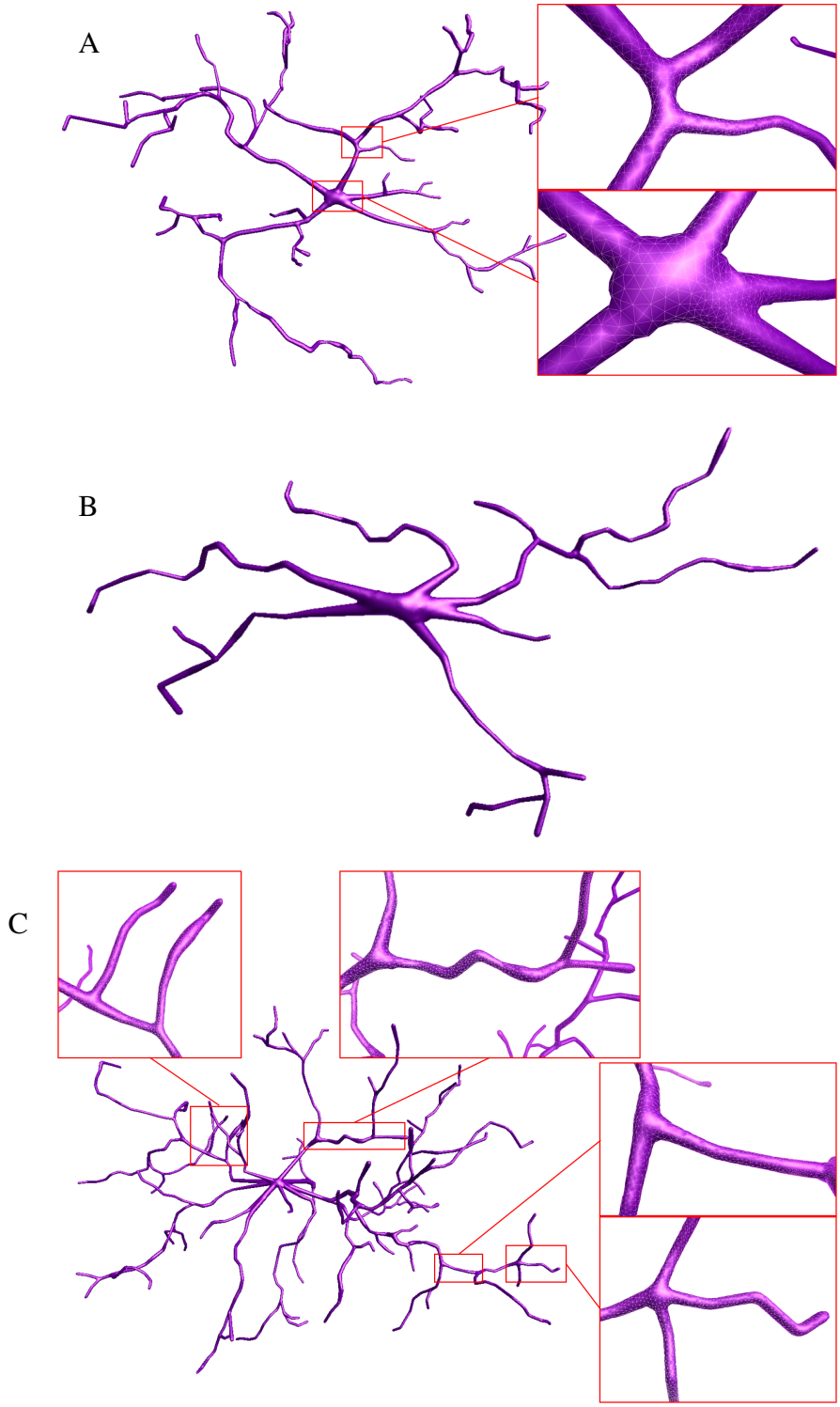

Figure 16. Some cells of Abdolhoseini_Kluge.

As the final mesh is progressively deformed from an initial sphere (Figure 18), no stitching operations are needed which usually introduce cracks or non-smoothness. In the dynamic mesh evolution, a quasi-uniform mesh structure is used to guide the edge split and collapse, and it does not allows too long or too short edges which are detrimental to a robust mesh generation.

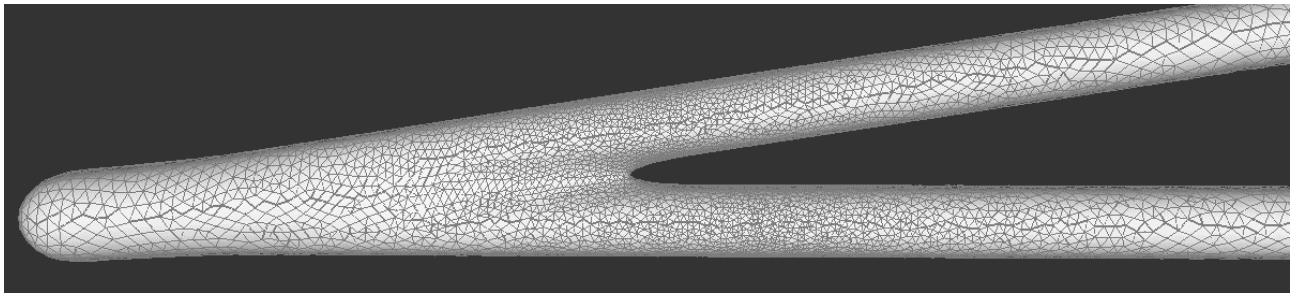

Figure 17. Adaptive vertex distribution at arborizations

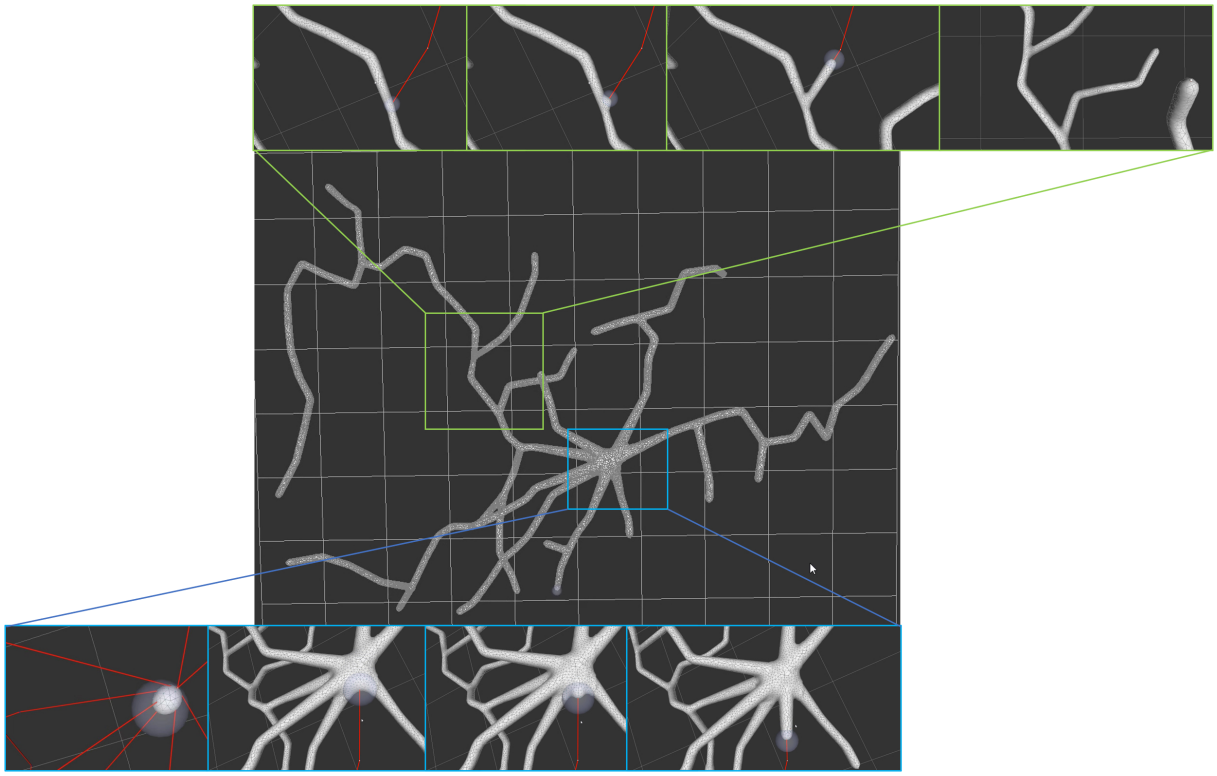

Figure 18. Progressive surface evolution

3.2 Comparison with Similar Tools

Neuronal membrane surface meshing is a fundamental preparation for neuronal electrical and chemical simulations. Earlier software packages, such as Neurolucida (Glaser and Glaser, 1990), NeuroConstruct (Gleeson et al., 2007), NeuGen (Eberhard et al., 2006), and Genesis (Wilson et al., 1988) provide approximations of neuron surfaces with mesh-based methods, but they tend to create meshes with low qualities, in which self-intersected parts usually occur in branching regions. Other methods such as the one presented in (Lasserre et al., 2011) starts from a sphere (made with quads) with a fixed resolution, and the dendrites are then generated by quad-extrusions starting from the soma. At the end of the method, a Catmull-Clark subdivision is employed to smooth the whole mesh, generating realistic, smooth, and closed meshes. However, the preservation of desirable properties of being 2D-manifold are not stated as an objective in the work.

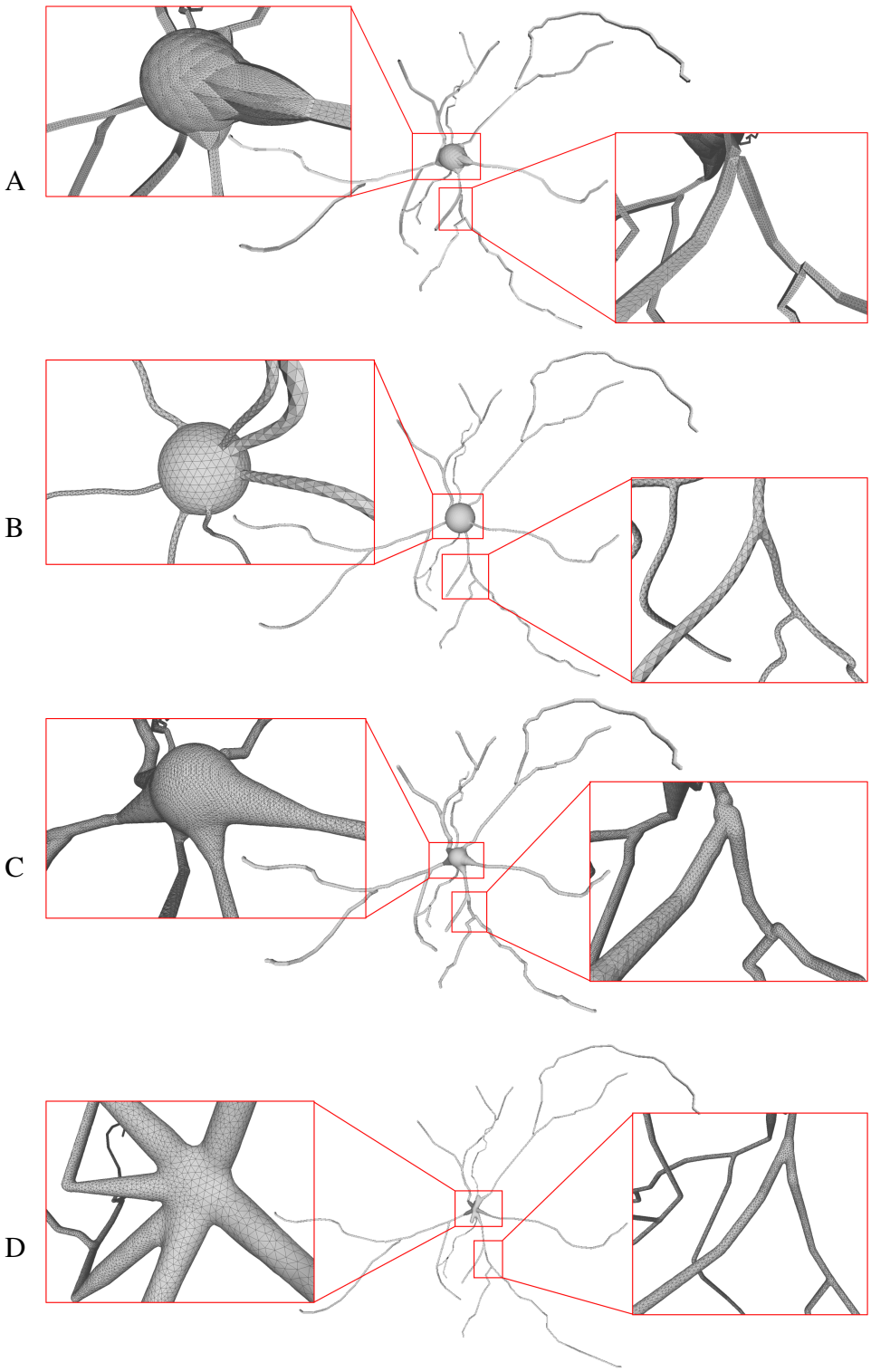

Figure 19. Comparison of the same neuron using different 3D representations. Models generated using NeuroTessMesh (A), AnaMorph (B), Neuromorphovis (C) and our proposed method (D).

292 **CTNG** (McDougal et al., 2013) uses constructive solid geometry to define a plausible reconstruction
293 without gaps, which represents a geometry as the unions and intersections of three-dimensional graphics
294 primitives. It then uses ”constructive cubes” to produce a watertight triangular mesh of the neuron surface.

However, it depends on the underlying Marching Cubes which create triangles with greatly varying aspect ratios.

Neuroize (Brito Menéndez et al., 2013) defines a physics-based mass-spring system to generate a neuron mesh, which could create a high-quality mesh. However, due to the versatility of the mass-spring system, complicated fine-tuning of several simulation parameters is required to achieve an accurate reconstruction.

NeuroTessMesh (Garcia-Cantero et al., 2017) improves the technique in Neuronize by applying an FEM (Finite Element Method) (Erleben et al., 2005) to simulate the deformation, which enables a convenient control over the mesh deformation. The approach approximates the cell bodies and the dendritic and axonal arbors in independent procedures that are later merged, resulting in a closed surface that approximates a whole neuron. However, NeuroTessMesh does not deal with mesh generation at branch points, and in particular, the junction between neurite and SOMA is not smooth, which is obviously not desirable as shown in Figure 19(A). Although NeuroTessMesh can adapt mesh resolution based on the distance to the camera, it cannot adapt mesh resolution based on different geometric details when generating individual neuron mesh models.

AnaMorph (Mörschel et al., 2017) uses a recursive tessellation algorithm starting from an icosahedron to construct the soma mesh as an initial mesh. The AnaMorph framework can produce relatively high-quality meshes, but it is not robust for building some complex neuronal structures, especially for high resolution modeling at branches. Moreover, a complex stitching operation has to be performed on adjacent branches in order to produce a water-tight surface (Figure 19(B)).

Neuromorphovis (Abdellah et al., 2018) extends an earlier meshing algorithm (Abdellah et al., 2017) capable of reconstructing piecewise watertight meshes that could be employed to visualize detailed electrophysiological activities obtained from voltage dynamics simulations. However, too many metaobjects have to be placed for creating a smooth-varying implicit field for extracting a neuronal membrane iso-surface. In addition, a post-optimization is required to improve the extracted triangles with greatly varying aspect ratios (Figure 19(C)).

3.3 Conclusion and Future Work

This paper presents a novel progressive approach for robustly generating high-quality 3D mesh neuron models based on widely used point-and-diameter input of morphological tracings, such as those available in public repositories NeuroMorpho.Org. The final mesh is created by iteratively evolving an initial soma along the neuron skeletons, which are represented in the form of point-and-diameter. The adopted skeletal mapping policy assigns each vertex of the mesh to a definite skeletal node, which enables an efficient query of ROI. Moreover, a half-edge structure of mesh representation in OpenMesh library is adopted to allows dynamic edge splits and collapses for adaptive vertex density distribution, as the half-edge structure is suitable for adjacent vertex/edge/facet queries. As described in the previous sections, a sphere with uniform distributed vertices is set as the initial soma for subsequent evolution, which performs dynamic local refinement, simplification and convolution surface approximation to generate a smooth neuron morphology with adaptive vertex density distribution. Therefore, the whole neuronal surface is always a closed 2D manifold mesh throughout all the evolution stages. Due to the summation property of the used convolution surfaces, the soma surface can be automatically created based on the embedded skeletons which are intersected with each other at the soma. Actually the by-product of bumps at branches assists to

create a plump shape without any special soma processing. In addition, to accelerate the computation-intensive convolution field generation, an analytical local convolution surface approximation is introduced to approximate the line-skeletons in the form of point-and-diameter.

Neural membrane surfaces with high-quality meshes and smooth branches can be robustly achieved using our approach, but the iterative deformation sacrifices the generation efficiency compared to a global mesh creation. Therefore, the advantageous superposition of convolution surfaces can be exploited to perform a "divide-and-conquer" policy for the whole neuron surface mesh generation in our future work. Moreover, a better soma can be modeled based on advanced physical simulation and contour-constrained implicit surface fitting, and it is worthwhile to extract a more precise soma model from raw volume data as our initial evolution mesh. Finally, our created mesh model is a 2D manifold surface which can be directly 3D printed for physical visualization and education purposes.

4 NOMENCLATURE

CONFLICT OF INTEREST STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

REFERENCES

Abdellah, M., Guerrero, N. R., Lapere, S., Coggan, J. S., Keller, D., Coste, B., et al. (2020). Interactive visualization and analysis of morphological skeletons of brain vasculature networks with vessmorphovis. *Bioinform.* 36, i534–i541. doi:10.1093/bioinformatics/btaa461

Abdellah, M., Hernando, J., Antille, N., Eilemann, S., Markram, H., and Schürmann, F. (2017). Reconstruction and visualization of large-scale volumetric models of neocortical circuits for physically-plausible in silico optical studies. *BMC bioinformatics* 18, 402

Abdellah, M., Hernando, J., Eilemann, S., Lapere, S., Antille, N., Markram, H., et al. (2018). Neuromorphovis: a collaborative framework for analysis and visualization of neuronal morphology skeletons reconstructed from microscopy stacks. *Bioinform.* 34, i574–i582. doi:10.1093/bioinformatics/bty231

Ahmed, A. G., Guo, J., Yan, D.-M., Franceschia, J.-Y., Zhang, X., and Deussen, O. (2016). A simple push-pull algorithm for blue-noise sampling. *IEEE transactions on visualization and computer graphics* 23, 2496–2508

Akkouche, S. and Galin, E. (2001). Adaptive implicit surface polygonization using marching triangles. In *Computer Graphics Forum* (Wiley Online Library), vol. 20, 67–80

Alliez, P., Ucelli, G., Gotsman, C., and Attene, M. (2008). Recent advances in remeshing of surfaces. *Shape analysis and structuring* , 53–82

Bloomenthal, J. (1994). An implicit surface polygonizer. *Graphics gems* 4, 324–350

Bloomenthal, J. and Shoemake, K. (1991). Convolution surfaces. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*. 251–256

Botsch, M. and Kobbelt, L. (2004). A remeshing approach to multiresolution modeling. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. 185–192

Botsch, M., Kobbelt, L., Pauly, M., Alliez, P., and Lévy, B. (2010). *Polygon mesh processing* (CRC press)

Bottino, A., Nuij, W., and van Overveld, C. (1996). How to shrinkwrap a critical point: an algorithm for the adaptive triangulation of iso-surfaces with arbitrary topology. In *Proceedings of IS96, the second Eurograhics/Siggraph workshop on Implicit Surfaces*. 73

Brito Menéndez, J. P., Mata Fernández, S., Bayona Beriso, S., Pastor Pérez, L., DeFelipe, J., and Benavides-Piccione, R. (2013). Neuronize: a tool for building realistic neuronal cell morphologies. *Frontiers in Neuroanatomy*

Carnevale, N. T. and Hines, M. L. (2006). *The NEURON book* (Cambridge University Press)

Cheng, S.-W., Dey, T. K., Shewchuk, J., and Sahni, S. (2013). *Delaunay mesh generation* (CRC Press Boca Raton)

Du, Q., Faber, V., and Gunzburger, M. (1999). Centroidal voronoi tessellations: Applications and algorithms. *SIAM review* 41, 637–676

Du, X., Liu, X., Yan, D.-M., Jiang, C., Ye, J., and Zhang, H. (2018). Field-aligned isotropic surface remeshing. In *Computer Graphics Forum* (Wiley Online Library), vol. 37, 343–357

Dunyach, M., Vanderhaeghe, D., Barthe, L., and Botsch, M. (2013). Adaptive remeshing for real-time mesh deformation. In *Eurographics 2013* (The Eurographics Association)

Ebeida, M. S., Rushdi, A. A., Awad, M. A., Mahmoud, A. H., Yan, D.-M., English, S. A., et al. (2016). Disk density tuning of a maximal random packing. In *Computer Graphics Forum* (Wiley Online Library), vol. 35, 259–269

Eberhard, J. P., Wanner, A., and Wittum, G. (2006). Neugen: a tool for the generation of realistic morphology of cortical neurons and neural networks in 3d. *Neurocomputing* 70, 327–342

Erleben, K., Sporring, J., Henriksen, K., and Dohlmann, H. (2005). *Physics-based animation* (Charles River Media Hingham)

Garcia-Cantero, J. J., Brito, J. P., Mata, S., Bayona, S., and Pastor, L. (2017). Neurotessmesh: A tool for the generation and visualization of neuron meshes and adaptive on-the-fly refinement. *Frontiers in Neuroinformatics* 11. doi:10.3389/fninf.2017.00038

Glaser, J. R. and Glaser, E. M. (1990). Neuron imaging with neurolucida-a pc-based system for image combining microscopy. *Computerized Medical Imaging and Graphics* 14, 307–317

Gleeson, P., Steuber, V., and Silver, R. A. (2007). neuroconstruct: a tool for modeling networks of neurons in 3d space. *Neuron* 54, 219–235

Hart, J. C. and Baker, B. (1996). Implicit modeling of tree surfaces. In *Proceedings of Implicit Surfaces' 96* (Citeseer), 143–152

Jin, X. and Tai, C.-L. (2002). Analytical methods for polynomial weighted convolution surfaces with various kernels. *Computers & Graphics* 26, 437–447

Jin, X., Tai, C.-L., and Zhang, H. (2009). Implicit modeling from polygon soup using convolution. *The Visual Computer* 25, 279–288

Kil, Y. J., Renzulli, P., Kreylos, O., Hamann, B., Monno, G., and Staadt, O. G. (2006). 3d warp brush modeling. *Computers & Graphics* 30, 610–618

Lasserre, S., Hernando, J., Hill, S., Schuermann, F., de Miguel Anasagasti, P., Abou Jaoudé, G., et al. (2011). A neuron membrane mesh representation for visualization of electrophysiological simulations. *IEEE Transactions on Visualization and Computer Graphics* 18, 214–227

Liu, Y.-J., Xu, C., Yi, R., Fan, D., and He, Y. (2016). Manifold differential evolution (mde): a global optimization method for geodesic centroidal voronoi tessellations on meshes. *ACM Trans. Graph.* 35, 243–1

Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive*

Techniques, SIGGRAPH 1987, Anaheim, California, USA, July 27-31, 1987, ed. M. C. Stone (ACM), 163–169. doi:10.1145/37401.37422

McCormack, J. and Sherstyuk, A. (1998). Creating and rendering convolution surfaces. In *Computer Graphics Forum* (Wiley Online Library), vol. 17, 113–120

McDougal, R. A., Hines, M. L., and Lytton, W. W. (2013). Water-tight membranes from neuronal morphology files. *Journal of Neuroscience Methods* 220, 167–178

Mörschel, K., Breit, M., and Queisser, G. (2017). Generating neuron geometries for detailed three-dimensional simulations using anamorph. *Neuroinformatics* 15, 247–269. doi:10.1007/s12021-017-9329-x

Stanculescu, L., Chaine, R., and Cani, M.-P. (2011). Freestyle: Sculpting meshes with self-adaptive topology. *Computers & Graphics* 35, 614–622

Van Overveld, K. and Wyvill, B. (2004). Shrinkwrap: An efficient adaptive algorithm for triangulating an iso-surface. *The visual computer* 20, 362–379

Vorsatz, J., Ro¨ ssl, C., and, and Seidel, H.-P. (2003). Dynamic remeshing and applications. *J. Comput. Inf. Sci. Eng.* 3, 338–344

Wang, Y., Yan, D.-M., Liu, X., Tang, C., Guo, J., Zhang, X., et al. (2018). Isotropic surface remeshing without large and small angles. *IEEE transactions on visualization and computer graphics* 25, 2430–2442

Wilson, M., Bhalla, U., Uhley, J., and Bower, J. (1988). Genesis: A system for simulating neural networks. *Advances in neural information processing systems* 1

Wyvill, G., McPheeters, C., and Wyvill, B. (1986). Soft objects. In *Advanced Computer Graphics* (Springer). 113–128

Yan, D.-M. and Wonka, P. (2015). Non-obtuse remeshing with centroidal voronoi tessellation. *IEEE transactions on visualization and computer graphics* 22, 2136–2144

Zhu, X., Guo, X., and Jin, X. (2013). Efficient polygonization of tree trunks modeled by convolution surfaces. *Science China Information Sciences* 56, 1–12