

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

Xiaoqiang Zhu¹, Xiaomei Liu¹, Sihu Liu¹, Yalan Shen², 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

2 ABSTRACT

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

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

1 INTRODUCTION

21 With a rapid technical development, increasing attention has been paid to biologically detailed brain
22 visualization and simulation. As the most significant information-processing cells in the brain, neurons
23 are made up of dendrites and axons, which are responsible for receiving and sending signals respectively.
24 Actually each neuron is an electrical entity and various methods have been proposed to simulate the
25 electrical behavior of neurons. However, due to the complexity of neuronal structure, most of these
26 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).

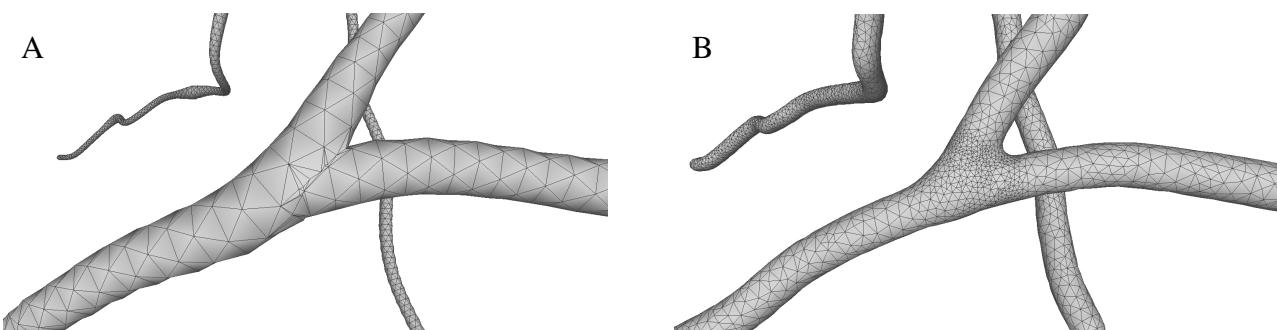


Figure 1. A neuron branch with coarse mesh vertices (**A**) and high-quality adaptive vertex density distributions (**B**).

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

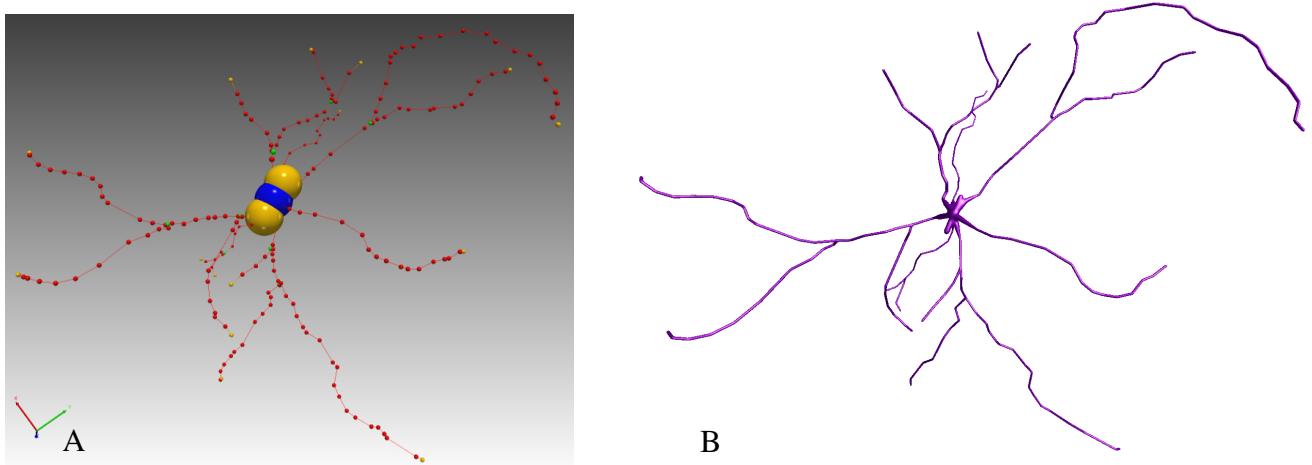


Figure 2. The Morphology of ANDERSON's basal ganglia. **(A)** The structure of ANDERSON's basal ganglia. **(B)** Full mesh.

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

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

2 METHOD

71 2.1 Architecture

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

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

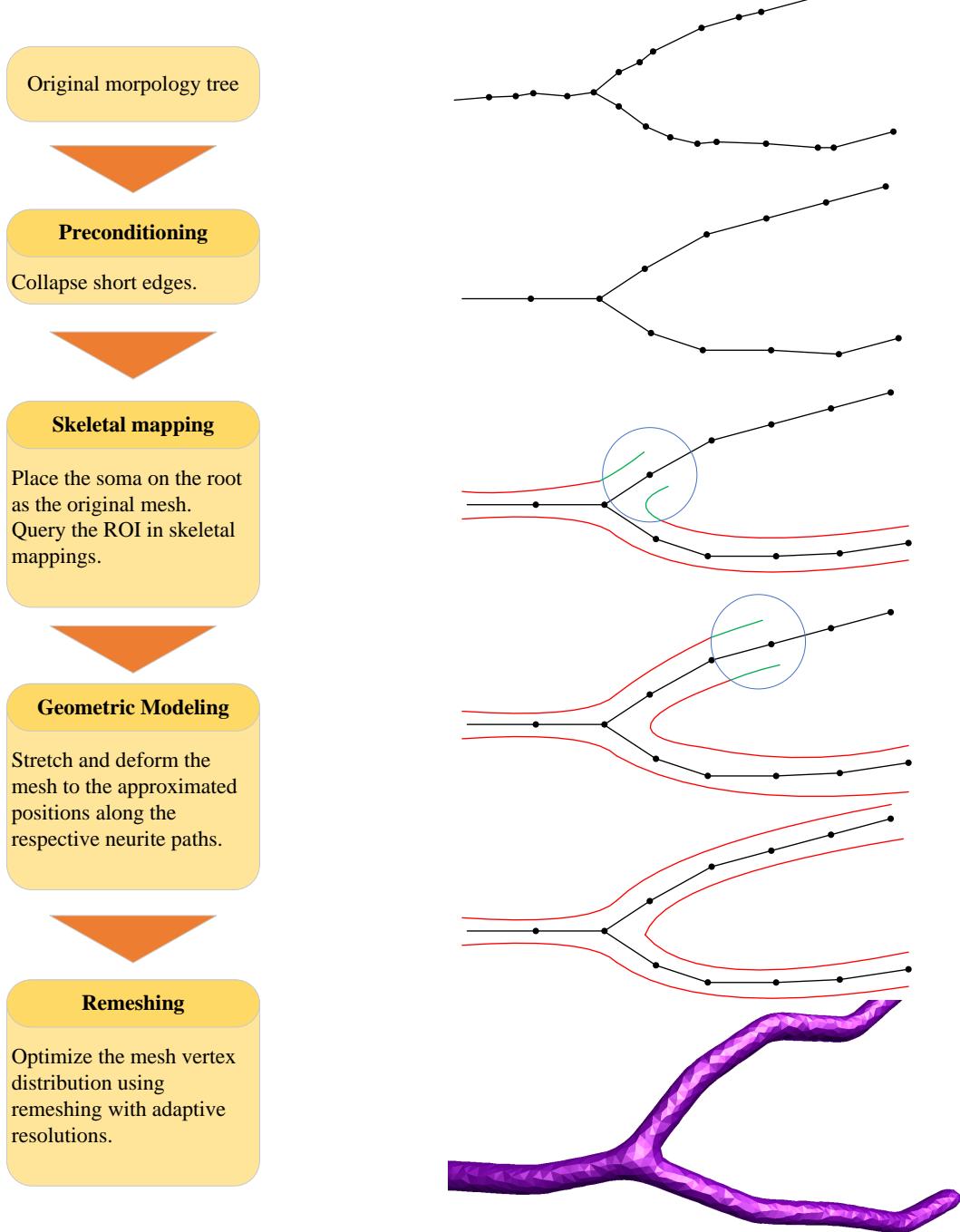


Figure 3. The progressive neuronal morphology modeling workflow. The procedure starts from an initial soma mesh at the root of the neuron tree, then searches the vertices affected by the skeleton-based implicit surfaces, and finally project these vertices onto the implicitly defined convolution surfaces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

83 2.2 Morphology Preconditioning

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

86 occur when the neurite segments are too short compared to the radii of their coincident skeleton vertices.
 87 They tend to create irregular meshes which are likely to be self-intersecting and may not be capable of
 88 2D manifold surface generation. However, there are indeed many samples with short neurite segments as
 89 well as generally irregular node distribution on NeuroMorpho.Org. Therefore the notion of short neurite
 90 segments are to be formalized using violation categories (cf. Figure 3).

91 2.3 ROI Query Strategy Based on Skeletal Mapping

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

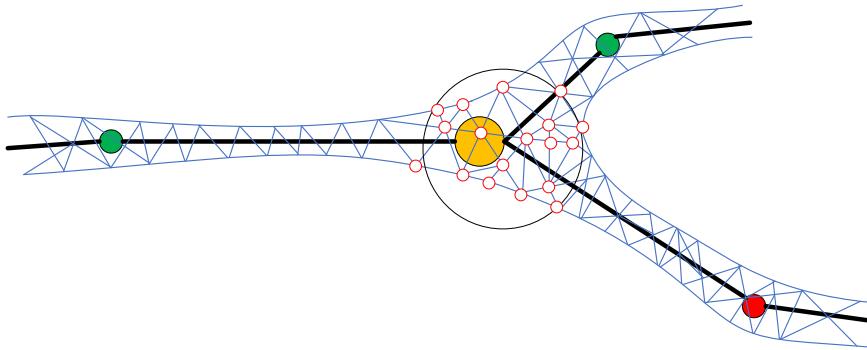


Figure 4. Skeletal mapping.

99 2.4 Local Convolution Surface Approximation

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

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

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

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

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

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

$$F(p) = \int_V g(q)f(p - q)dV = (f \otimes g)(p) \quad (2)$$

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

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

130 where R is the effective radius of the kernel function.

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

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

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

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

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

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

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

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

146 The implementation of approximation is to project each vertex of the remeshed surface onto the implicit
 147 convolution surface in their respective gradient directions. The evolution step length of the vertex p can
 148 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
 149 projection step length of a new vertex can be derived from its adjacent vertices (Figure 5).

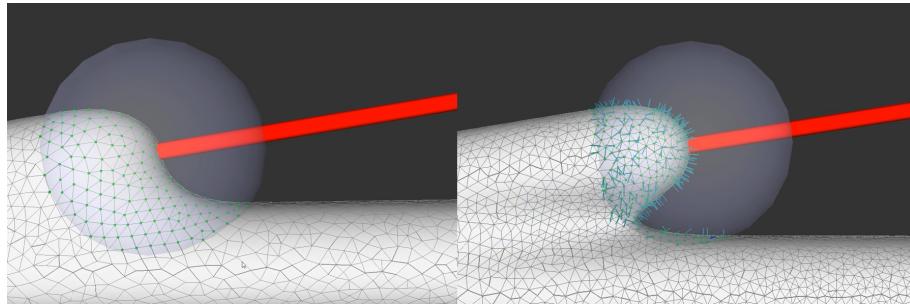


Figure 5. Convolution surface approximation at branches using progressive evolution.

150 2.5 Remeshing Algorithm based on Quasi-uniform Mesh Structure

151 Isotropic remeshing is a valid scheme to optimize the distribution of mesh vertices and topological
 152 connectivity, and the surveys in (Botsch et al., 2010; Alliez et al., 2008) have proposed various methods
 153 for surface remeshing. Numerous techniques like the particle-based methods (Ahmed et al., 2016), the
 154 Delaunay refinement methods (Cheng et al., 2013), and the randomized sampling methods (Ebeida et al.,
 155 2016) pay attention to generating isotropic meshes. The methods based on Centroidal Voronoi (Du et al.,
 156 1999) tessellation attempt to make each point in accord with the centroid of its Voronoi region. There
 157 are various ways to calculate a Voronoi diagram on surfaces, such as mesh parameterizations, discrete
 158 clustering, restricted Voronoi diagram (Yan and Wonka, 2015), and geodesic Voronoi diagram (Liu et al.,
 159 2016). The fundamental theory of the field-based methods is that isotropic triangular meshes can be
 160 taken out from sixway rotational symmetry directional fields (Du et al., 2018). In addition, there are
 161 edge-based approaches are based on local operators, including edge split, edge collapse, edge flip, and
 162 vertex repositioning (Botsch and Kobbelt, 2004; Dunyach et al., 2013; Wang et al., 2018).

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

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

Algorithm Remeshing

Input: *init_surface, target_edge_length*

Output: quasi-uniform mesh

```

1: function REMESH(init_surface, target_edge_length)
2:   result  $\leftarrow$  init_surface
3:   low  $\leftarrow \alpha * \text{target\_edge\_length}$ 
4:   high  $\leftarrow \beta * \text{target\_edge\_length}$ 
5:   for vHandle : vHandles do
6:     split_long_edges(high)
7:     collapse_short_edges(low, high)
8:     equalize_valences()
9:     tangential_relaxation()
10:    project_to_surface()
11:   end for
12:   return result
13: end function

```

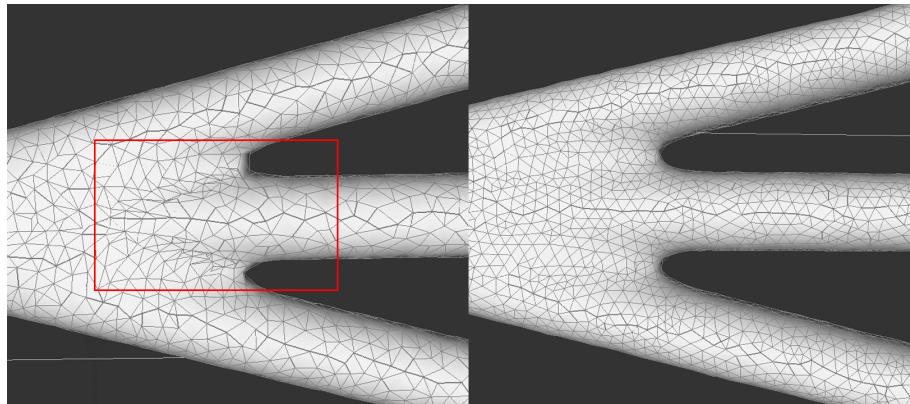


Figure 6. An optimized remeshing based on quasi-uniform mesh.

173 2.5.1 Quasi-uniform Mesh Configuration

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

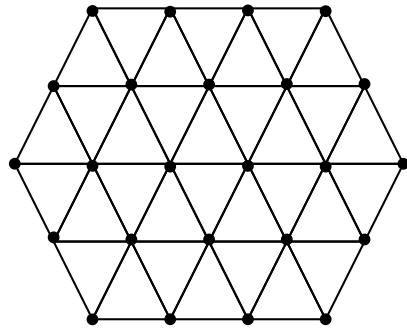


Figure 7. A uniform mesh.

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

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

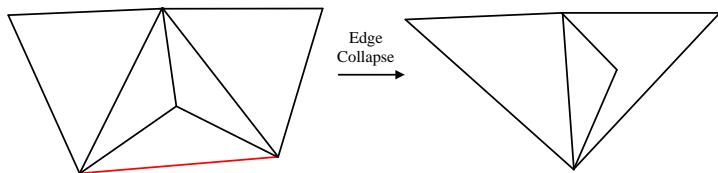


Figure 8. Illegal collapse.

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

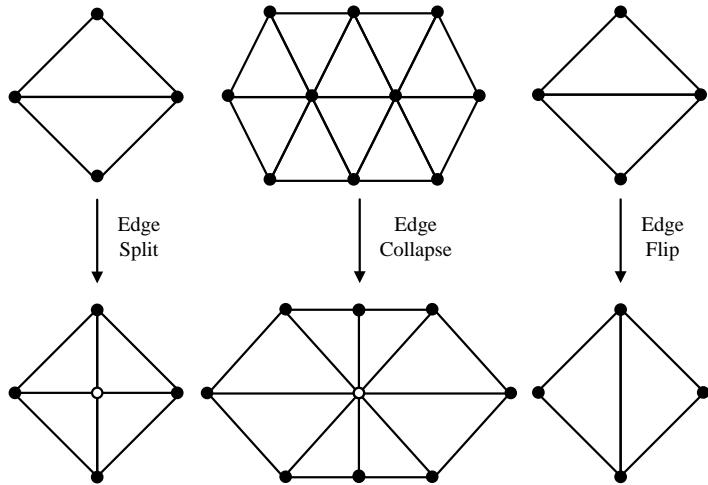


Figure 9. Edge split, collapse and flip.

201 2.5.2 Area-equalizing Remeshing

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

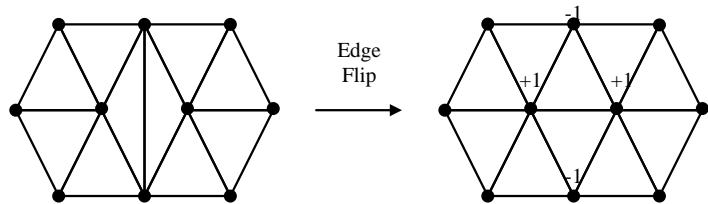
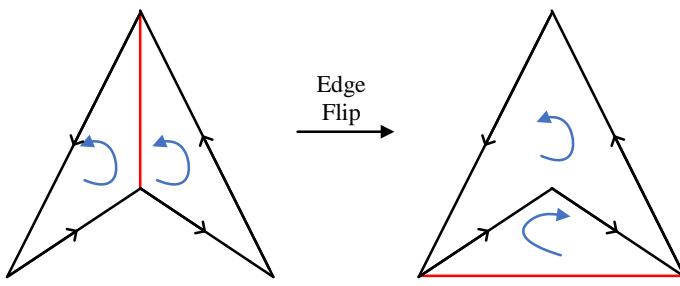
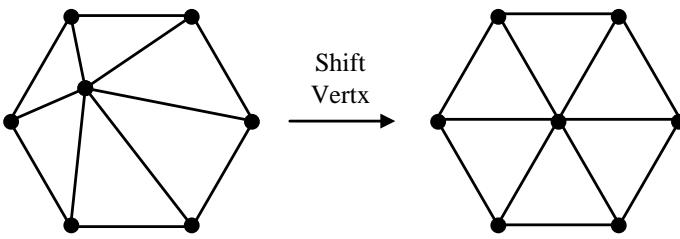


Figure 10. Valence equalization

211 After iteratively performing the above optimization operations on the mesh M , we can get a relatively
 212 uniform triangular mesh. The edge lengths of the mesh are basically close to the target length l , which
 213 is between d and l_T , and the valences of the vertices are close to 6. However, the triangle areas, the edge
 214 lengths and the vertex valances may still vary across the mesh surface, which can be further optimized.
 215 Since k -valence refers to a vertex coincident to k triangles with different areas. Therefore, the averaged
 216 area of each vertex contributed by its coincident triangles is dependent of these triangle geometries. To
 217 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.

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

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

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

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

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

227 2.6 Strategies for Adaptive Mesh Vertex Density

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

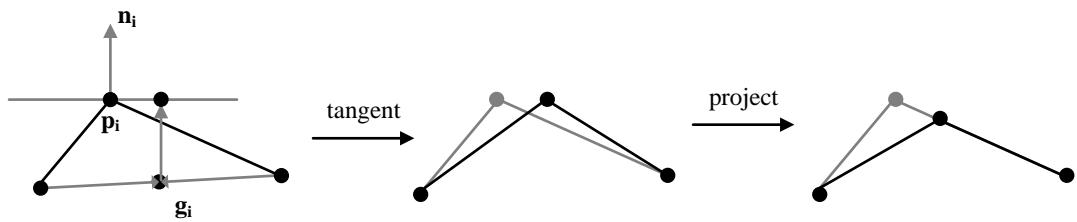


Figure 13. Tangential relaxation and vertex projection.

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

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

242 1. **Separating Point.** According to the angle of a neuron branch and the support radius R of the target
 243 convolution surfaces, the separating position t at the branch can be calculated. Since the distances
 244 between t and the skeleton segments of the both branches are R , the projection points of t on the
 245 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).

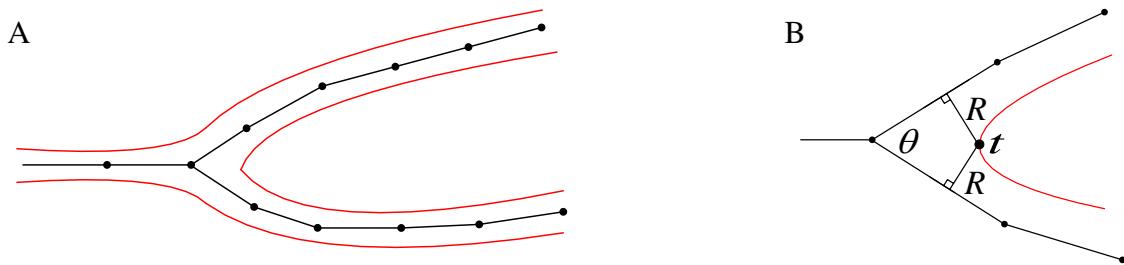


Figure 14. Adaptive mesh resolutions at branches. (A) presents a branch, and (B) shows the separating position t .

246
 247 2. **Maximum Vertex Density.** As the neuron morphology at the branching t usually creates a surface
 248 with the locally highest curvature, the maximum vertex densities are assigned for the mesh in the
 249 nearby region. Therefore, triangles with the minimum edge length are generated in the separating
 250 region based on a marching along the skeleton with the minimum step size.

251 3. **Minimum Vertex Density.** On the other hand, the maximum target edge length can be usually set as
 252 the average edge length of the initial mesh, as our mesh evolution begins with an initial soma which

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

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

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

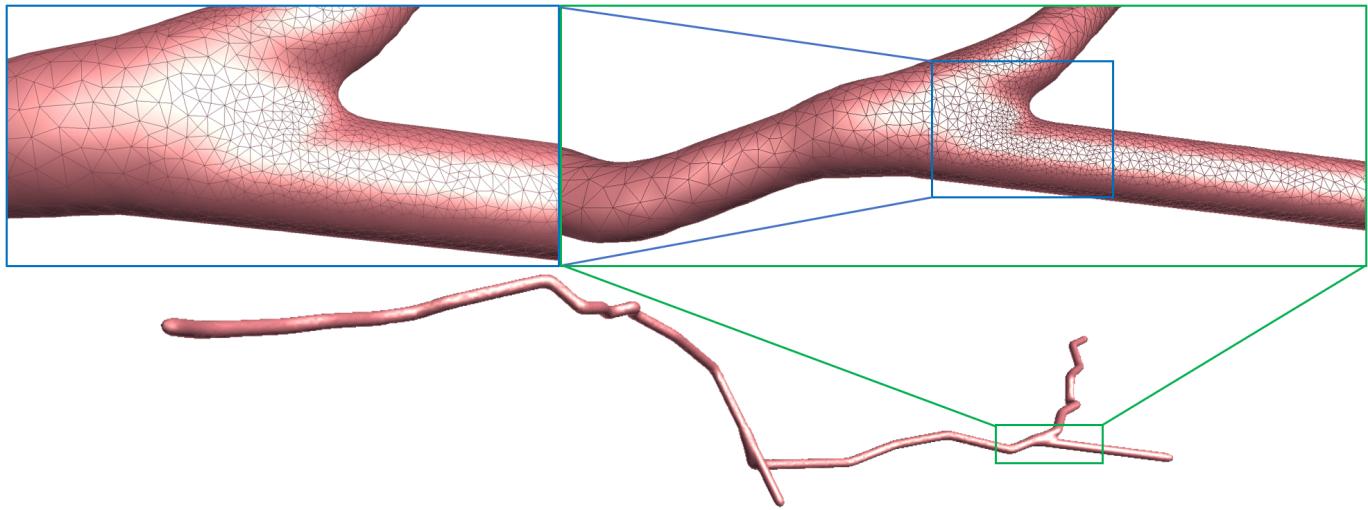


Figure 15. Adaptive mesh vertex distributions in a separating region.

3 RESULTS AND DISCUSSION

260 3.1 Results

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

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

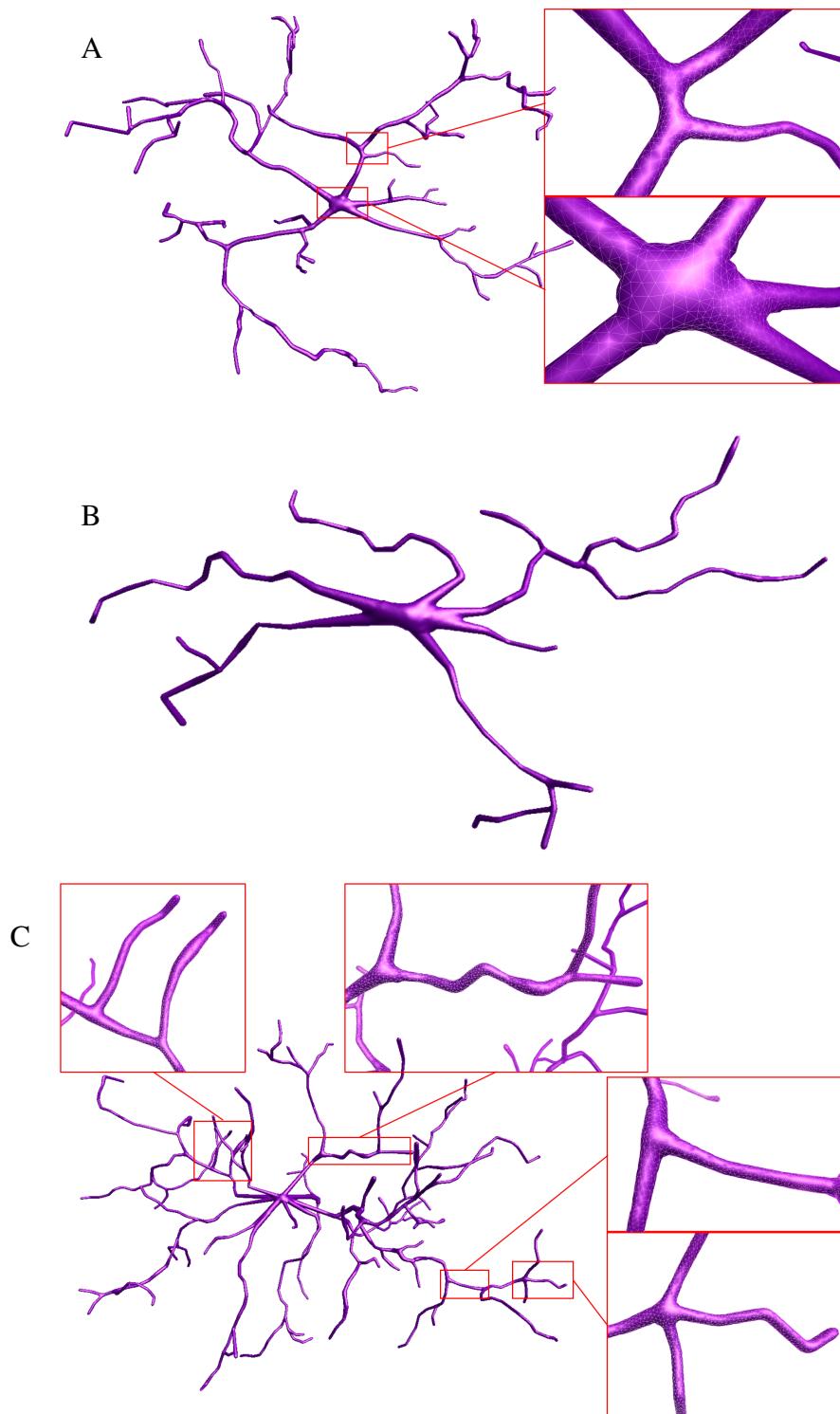


Figure 16. Some cells of Abdolhoseini_Kluge.

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

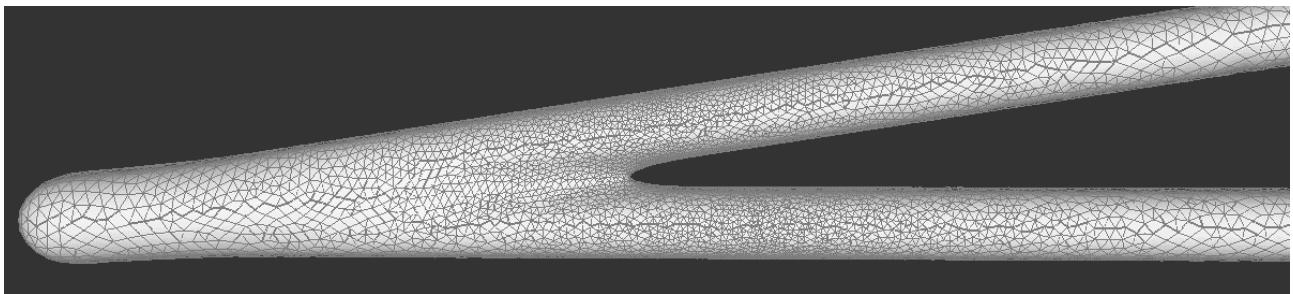


Figure 17. Adaptive vertex distribution at arborizations

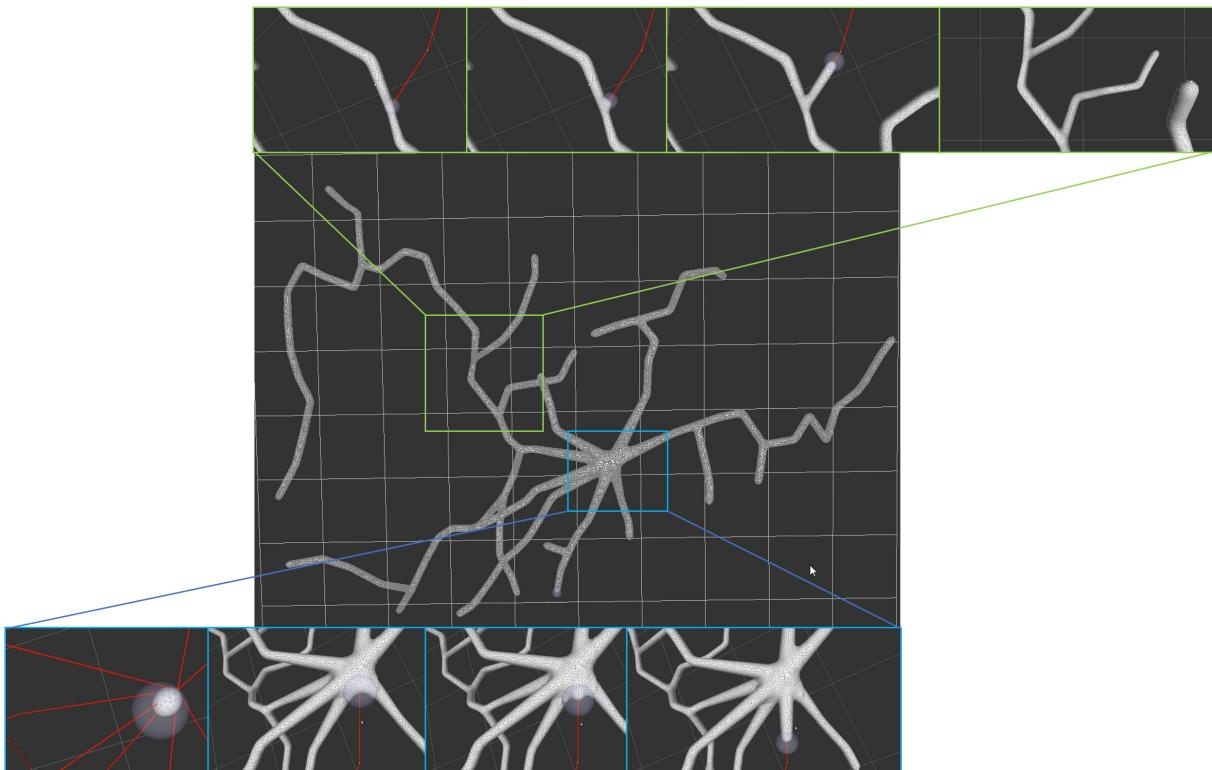


Figure 18. Progressive surface evolution

281 3.2 Comparison with Similar Tools

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

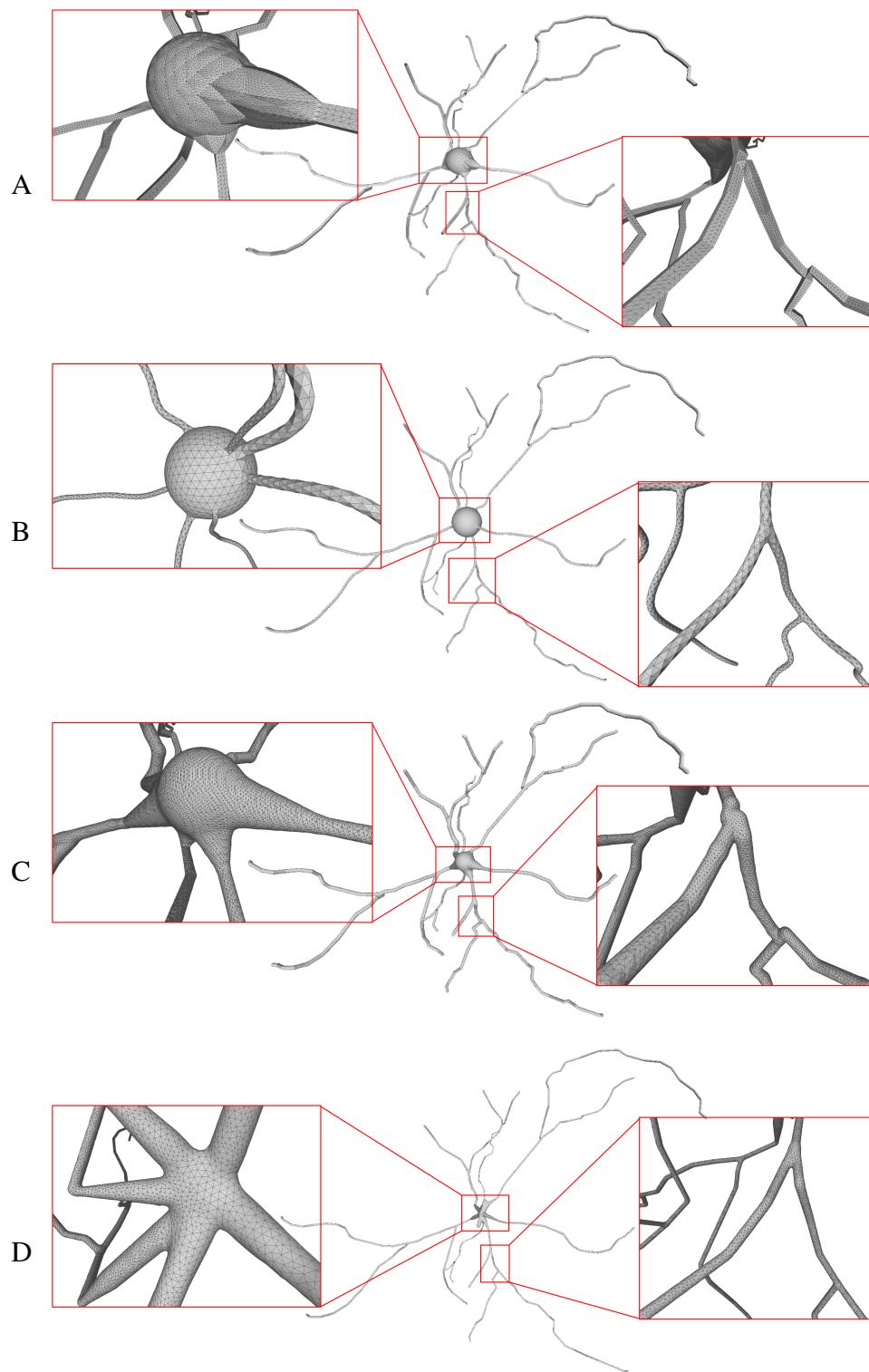


Figure 19. Comparison of the same neuron using different 3D representations. Models generated using NeuroTessMesh (A), AnaMorph (B), Neuromorphovis (C) and our proposed method (D).

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.

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

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

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

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

314 **Neuromorphovis** (Abdellah et al., 2018) extends an earlier meshing algorithm (Abdellah et al.,
315 2017) capable of reconstructing piecewise watertight meshes that could be employed to visualize
316 detailed electrophysiological activities obtained from voltage dynamics simulations. However, too many
317 metaobjects have to be placed for creating a smooth-varying implicit field for extracting a neuronal
318 membrane iso-surface. In addition, a post-optimization is required to improve the extracted triangles with
319 greatly varying aspect ratios (Figure 19(C)).

320 3.3 Conclusion and Future Work

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

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

338 Neural membrane surfaces with high-quality meshes and smooth branches can be robustly achieved
339 using our approach, but the iterative deformation sacrifices the generation efficiency compared to a global
340 mesh creation. Therefore, the advantageous superposition of convolution surfaces can be exploited to
341 perform a “divide-and-conquer” policy for the whole neuron surface mesh generation in our future work.
342 Moreover, a better soma can be modeled based on advanced physical simulation and contour-constrained
343 implicit surface fitting, and it is worthwhile to extract a more precise soma model from raw volume data as
344 our initial evolution mesh. Finally, our created mesh model is a 2D manifold surface which can be directly
345 3D printed for physical visualization and education purposes.

CONFLICT OF INTEREST STATEMENT

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

REFERENCES

- 348 Abdellah, M., Guerrero, N. R., Lapere, S., Coggan, J. S., Keller, D., Coste, B., et al. (2020). Interactive
349 visualization and analysis of morphological skeletons of brain vasculature networks with vessmorphovis.
350 *Bioinform.* 36, i534–i541. doi:10.1093/bioinformatics/btaa461
- 351 Abdellah, M., Hernando, J., Antille, N., Eilemann, S., Markram, H., and Schürmann, F. (2017).
352 Reconstruction and visualization of large-scale volumetric models of neocortical circuits for
353 physically-plausible in silico optical studies. *BMC bioinformatics* 18, 402
- 354 Abdellah, M., Hernando, J., Eilemann, S., Lapere, S., Antille, N., Markram, H., et al. (2018).
355 Neuromorphovis: a collaborative framework for analysis and visualization of neuronal morphology
356 skeletons reconstructed from microscopy stacks. *Bioinform.* 34, i574–i582. doi:10.1093/bioinformatics/
357 bty231
- 358 Ahmed, A. G., Guo, J., Yan, D.-M., Franceschia, J.-Y., Zhang, X., and Deussen, O. (2016). A simple
359 push-pull algorithm for blue-noise sampling. *IEEE transactions on visualization and computer graphics*
360 23, 2496–2508
- 361 Akkouche, S. and Galin, E. (2001). Adaptive implicit surface polygonization using marching triangles. In
362 *Computer Graphics Forum* (Wiley Online Library), vol. 20, 67–80
- 363 Alliez, P., Ucelli, G., Gotsman, C., and Attene, M. (2008). Recent advances in remeshing of surfaces.
364 *Shape analysis and structuring*, 53–82
- 365 Bloomenthal, J. (1994). An implicit surface polygonizer. *Graphics gems* 4, 324–350
- 366 Bloomenthal, J. and Shoemake, K. (1991). Convolution surfaces. In *Proceedings of the 18th annual
367 conference on Computer graphics and interactive techniques*. 251–256
- 368 Botsch, M. and Kobbel, L. (2004). A remeshing approach to multiresolution modeling. In *Proceedings of
369 the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. 185–192
- 370 Botsch, M., Kobbel, L., Pauly, M., Alliez, P., and Lévy, B. (2010). *Polygon mesh processing* (CRC press)
- 371 Bottino, A., Nuij, W., and van Overveld, C. (1996). How to shrinkwrap a critical point: an algorithm for
372 the adaptive triangulation of iso-surfaces with arbitrary topology. In *Proceedings of IS96, the second
373 Eurographics/Siggraph workshop on Implicit Surfaces*. 73

- 374 Brito Menéndez, J. P., Mata Fernández, S., Bayona Beriso, S., Pastor Pérez, L., DeFelipe, J., and
375 Benavides-Piccione, R. (2013). Neuronize: a tool for building realistic neuronal cell morphologies.
376 *Frontiers in Neuroanatomy*
- 377 Carnevale, N. T. and Hines, M. L. (2006). *The NEURON book* (Cambridge University Press)
- 378 Cheng, S.-W., Dey, T. K., Shewchuk, J., and Sahni, S. (2013). *Delaunay mesh generation* (CRC Press
379 Boca Raton)
- 380 Du, Q., Faber, V., and Gunzburger, M. (1999). Centroidal voronoi tessellations: Applications and
381 algorithms. *SIAM review* 41, 637–676
- 382 Du, X., Liu, X., Yan, D.-M., Jiang, C., Ye, J., and Zhang, H. (2018). Field-aligned isotropic surface
383 remeshing. In *Computer Graphics Forum* (Wiley Online Library), vol. 37, 343–357
- 384 Dunyach, M., Vanderhaeghe, D., Barthe, L., and Botsch, M. (2013). Adaptive remeshing for real-time
385 mesh deformation. In *Eurographics 2013* (The Eurographics Association)
- 386 Ebeida, M. S., Rushdi, A. A., Awad, M. A., Mahmoud, A. H., Yan, D.-M., English, S. A., et al. (2016).
387 Disk density tuning of a maximal random packing. In *Computer Graphics Forum* (Wiley Online
388 Library), vol. 35, 259–269
- 389 Eberhard, J. P., Wanner, A., and Wittum, G. (2006). Neugen: a tool for the generation of realistic
390 morphology of cortical neurons and neural networks in 3d. *Neurocomputing* 70, 327–342
- 391 Erleben, K., Sporring, J., Henriksen, K., and Dohlmann, H. (2005). *Physics-based animation* (Charles
392 River Media Hingham)
- 393 Garcia-Cantero, J. J., Brito, J. P., Mata, S., Bayona, S., and Pastor, L. (2017). Neurotessmesh: A tool
394 for the generation and visualization of neuron meshes and adaptive on-the-fly refinement. *Frontiers in
395 Neuroinformatics* 11. doi:10.3389/fninf.2017.00038
- 396 Glaser, J. R. and Glaser, E. M. (1990). Neuron imaging with neurolucida-a pc-based system for image
397 combining microscopy. *Computerized Medical Imaging and Graphics* 14, 307–317
- 398 Gleeson, P., Steuber, V., and Silver, R. A. (2007). neuroconstruct: a tool for modeling networks of neurons
399 in 3d space. *Neuron* 54, 219–235
- 400 Hart, J. C. and Baker, B. (1996). Implicit modeling of tree surfaces. In *Proceedings of Implicit Surfaces'*
401 96 (Citeseer), 143–152
- 402 Jin, X. and Tai, C.-L. (2002). Analytical methods for polynomial weighted convolution surfaces with
403 various kernels. *Computers & Graphics* 26, 437–447
- 404 Jin, X., Tai, C.-L., and Zhang, H. (2009). Implicit modeling from polygon soup using convolution. *The
405 Visual Computer* 25, 279–288
- 406 Kil, Y. J., Renzulli, P., Kreylos, O., Hamann, B., Monno, G., and Staadt, O. G. (2006). 3d warp brush
407 modeling. *Computers & Graphics* 30, 610–618
- 408 Lasserre, S., Hernando, J., Hill, S., Schuermann, F., de Miguel Anasagasti, P., Abou Jaoudé, G., et al.
409 (2011). A neuron membrane mesh representation for visualization of electrophysiological simulations.
410 *IEEE Transactions on Visualization and Computer Graphics* 18, 214–227
- 411 Liu, Y.-J., Xu, C., Yi, R., Fan, D., and He, Y. (2016). Manifold differential evolution (mde): a global
412 optimization method for geodesic centroidal voronoi tessellations on meshes. *ACM Trans. Graph.* 35,
413 243–1
- 414 Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction
415 algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive
416 Techniques, SIGGRAPH 1987, Anaheim, California, USA, July 27-31, 1987*, ed. M. C. Stone (ACM),
417 163–169. doi:10.1145/37401.37422

- 418 McCormack, J. and Sherstyuk, A. (1998). Creating and rendering convolution surfaces. In *Computer*
419 *Graphics Forum* (Wiley Online Library), vol. 17, 113–120
- 420 McDougal, R. A., Hines, M. L., and Lytton, W. W. (2013). Water-tight membranes from neuronal
421 morphology files. *Journal of Neuroscience Methods* 220, 167–178
- 422 Mörschel, K., Breit, M., and Queisser, G. (2017). Generating neuron geometries for detailed
423 three-dimensional simulations using anamorph. *Neuroinformatics* 15, 247–269. doi:10.1007/
424 s12021-017-9329-x
- 425 Stanculescu, L., Chaine, R., and Cani, M.-P. (2011). Freestyle: Sculpting meshes with self-adaptive
426 topology. *Computers & Graphics* 35, 614–622
- 427 Van Overveld, K. and Wyvill, B. (2004). Shrinkwrap: An efficient adaptive algorithm for triangulating an
428 iso-surface. *The visual computer* 20, 362–379
- 429 Vorsatz, J., Roß, C., and Seidel, H.-P. (2003). Dynamic remeshing and applications. *J. Comput.*
430 *Inf. Sci. Eng.* 3, 338–344
- 431 Wang, Y., Yan, D.-M., Liu, X., Tang, C., Guo, J., Zhang, X., et al. (2018). Isotropic surface remeshing
432 without large and small angles. *IEEE transactions on visualization and computer graphics* 25, 2430–
433 2442
- 434 Wilson, M., Bhalla, U., Uhley, J., and Bower, J. (1988). Genesis: A system for simulating neural networks.
435 *Advances in neural information processing systems* 1
- 436 Wyvill, G., McPheevers, C., and Wyvill, B. (1986). Soft objects. In *Advanced Computer Graphics*
437 (Springer). 113–128
- 438 Yan, D.-M. and Wonka, P. (2015). Non-obtuse remeshing with centroidal voronoi tessellation. *IEEE*
439 *transactions on visualization and computer graphics* 22, 2136–2144
- 440 Zhu, X., Guo, X., and Jin, X. (2013). Efficient polygonization of tree trunks modeled by convolution
441 surfaces. *Science China Information Sciences* 56, 1–12

442 **Appendices**

A RESPONSE TO REVIEWERS' COMMENTS

443 First of all, we would like to thank the anonymous reviewers for their constructive and valuable comments,
444 which will certainly help us improve the manuscript. In what follows, we will list the changes colored in
445 **BLUE** in our revised version and provide further explanations in relation to the reviewers' comments.

446 **A.1 Reviewer #1:**

447 The authors present a meshing technique to create watertight polygonal surface meshes from neuronal
448 morphology skeletons (in point-and-diameter format). The technique is similar to using implicit geometry
449 or meta-balls to convolve a spherical field along the center-lines of the morphology skeleton to construct
450 the facets of the membrane. A local mapping technique is adopted to update the membrane vertices within
451 a small ROI to minimize the branching artifacts and to build smooth surface. Mesh tessellation is adjusted
452 by some quasi-uniform rules and based on the surface curvature of the mesh taking into account the
453 morphological descriptions.

454 The paper has a good flow, and is indeed informative. But it cannot be accepted in the current form due
455 to the following issues:

456 **Q101:** Section 4 Nomenclature is EMPTY!

457 **A101:** Thanks for the suggestions. We have deleted the empty Nomenclature section.

458

459 **Q102:** The robustness can be either mathematically proven or addressed based on trying the meshing
460 algorithm on thousands of neuronal morphologies.

461 **A102:** Thanks for the suggestions. We have done Please search "A102" for details in Section ?? of
462 the revised paper.

463

464 **Q103:** The authors mention that their algorithm is high quality. On which basis this statement is made?
465 Are there any qualitative analysis with which the quality is assessed or compared to other techniques? If
466 yes, please add them to the results section.

467 **A103:** Thanks for the suggestions. We have done Please search "A103" for details in Section ?? of
468 the revised paper.

469

470 **Q104:** On-the-fly could mean interactive mesh generation, however, I don't find any performance
471 evaluation section in the paper.

472 **A104:** Thanks for the suggestions. We have done Please search "A104" for details in Section ?? of
473 the revised paper.

474

475 **Q105:** In the morphology pre-conditions, what are the changes applied on the morphology? Does the
476 structure change?

477 **A105:** Thanks for the suggestions. We have done Please search “A105” for details in Section ?? of
478 the revised paper.

479

480 **Q106:** Is it guaranteed that the remeshing and adaptive vertex mapping algorithm preserve the
481 watertightness of the mesh?

482 **A106:** Thanks for the suggestions. We have done Please search “A106” for details in Section ?? of
483 the revised paper.

484

485 **Q107:** Mesh quality assessment must be added.

486 **A107:** Thanks for the suggestions. We have done Please search “A107” for details in Section ?? of
487 the revised paper.

488

489 **Q108:** The current results are insufficient. The algorithm must be tested at least with a group of 10-25
490 neurons with different morphological types and extracted from multiple brain regions.

491 **A108:** Thanks for the suggestions. We have done Please search “A108” for details in Section ?? of
492 the revised paper.

493

494 **Q109:** The details of the implementation are not given? In which language the code is written? What are
495 the dependencies of the code? Is the code going to be open sourced and available to public users or not?

496 **A109:** Thanks for the suggestions. We have done Please search “A109” for details in Section ?? of
497 the revised paper.

498

499 **Q110:** The author must provide complete quality analysis on the resulting meshes (refer to <https://coreform.com/papers/verdictqualitylibrary.pdf>) including: 1) distributions of the edge-ratio,
500 triangle area, and edge-to-radius ratio; 2) watertightness verification; 3) measurements of the relative error
501 between the surface of the resulting mesh and the morphological extent from the morphology file.

503 **A110:** Thanks for the suggestions. We have done Please search “A110” for details in Section ?? of
504 the revised paper.

505

506 **Q111:** Can the authors append a list of created meshes with their algorithm to test whether they are
507 watertight or not?

508 **A111:** Thanks for the suggestions. We have done Please search “A111” for details in Section ?? of
509 the revised paper.

510

511 **Q112:** Line 24: Actually should be removed.

512 **A112:** Thanks for the suggestions. We have removed ”Actually”. Please search “A112” for details in
513 Section 1 of the revised paper.

514

515 **Q113:** Line 27: Bad language construct: in a high fidelity dimension?

516 **A113:** Thanks for the suggestions. We have done Please search “A113” for details in Section ?? of
517 the revised paper.

518

519 **Q114:** Line 30: chemical-electrical interactions ? for reaction-diffusion simulations.

520 **A114:** Thanks for the suggestions. We have done Please search “A114” for details in Section ?? of
521 the revised paper.

522

523 **Q115:** Line 32: from microscope images ? from microscopy images.

524 **A115:** Thanks for the suggestions. We have done Please search “A115” for details in Section ?? of
525 the revised paper.

526

527 **Q116:** Line 40: to use the traced structures for problems ? to use the morphologies per se in visualization
528 or simulation.

529 **A116:** Thanks for the suggestions. We have done Please search “A116” for details in Section ?? of
530 the revised paper.

531

532 **Q117:** Line 44: complicated region ? complicated part or process.

533 **A117:** Thanks for the suggestions. We have done Please search “A117” for details in Section ?? of
534 the revised paper.

535

536 **Q118:** Line 56: to derive iterative deformations of the initial triangular mesh (this needs further
537 explanation)

538 **A118:** Thanks for the suggestions. We have done Please search “A118” for details in Section ?? of
539 the revised paper.

540

541 **Q119:** Line 58: robustly (see the issues below)

542 **A119:** Thanks for the suggestions. We have done Please search “A119” for details in Section ?? of
543 the revised paper.

544

545 **Q120:** Line 59: The entire statement is wrong, because stochastic reaction-diffusion simulation use
546 domains in the form of tetra- or quad- domains, therefore tetrahedral or hexahedral meshes are required.

547 **A120:** Thanks for the suggestions. We have done Please search “A120” for details in Section ?? of
548 the revised paper.

549

550 **Q121:** Line 74: what is cf. ?

551 **A121:** Thanks for the suggestions. We have done Please search “A121” for details in Section ?? of
552 the revised paper.

553

554 **Q122:** In Figure 3 flow chart, what is a short edge? On which basis the points (or samples) are removed
555 from the morphology skeleton?

556 **A122:** Thanks for the suggestions. We have done Please search “A122” for details in Section ?? of
557 the revised paper.

558

559 **Q123:** Line 85: the SOMA should be in lower case (soma), why it is capitalized?

560 **A123:** Thanks for the suggestions. We have done Please search “A123” for details in Section ?? of
561 the revised paper.

562

563 **Q124:** Line 85: awful must not be used in scientific or technical papers! The paper must be reviewed by
564 a native or professional English editor.

565 **A124:** Thanks for the suggestions. We have done Please search “A124” for details in Section ?? of
566 the revised paper.

567

568 **Q125:** Line 89: A reference to NeuroMorpho.Org must be added. (Ascoli et al.)

569 **A125:** Thanks for the suggestions. We have done Please search “A125” for details in Section ?? of
570 the revised paper.

571

572 **Q126:** Line 90: what is cf. ? The authors can use: see Figure 3 or refer to Figure 3 instead.

573 **A126:** Thanks for the suggestions. We have done Please search “A126” for details in Section ?? of
574 the revised paper.

575

576 **Q127:** Line 92: strategy based ? strategy-based

577 **A127:** Thanks for the suggestions. We have done Please search “A127” for details in Section ?? of
578 the revised paper.

579

580 **Q128:** Line 109: Abdellah et al. 2020 create mesh models of blood vessels not morphological models!

581 **A128:** Thanks for the suggestions. We have done Please search “A128” for details in Section ?? of
582 the revised paper.

583

584 **Q129:** Line 110: The astrocytic mesh generation is presented in Abdellah et al. 2021(https://academic.oup.com/bioinformatics/article/37/Supplement_1/i426/6319688) not in Abdellah et al.
585 2020.

587 **A129:** Thanks for the suggestions. We have done Please search “A129” for details in Section ?? of
588 the revised paper.

589

590 **Q130:** Line 113: SOMA is just the cell body, why it is always CAPITALIZED?

591 **A130:** Thanks for the suggestions. We have done Please search “A130” for details in Section ?? of
592 the revised paper.

593

594 **Q131:** Line 134: What is a morphological mesh? The morphology is the point-and-diameter
595 representation, while the mesh is the faces and vertices reflecting the membrane. Therefore the term is
596 confusing.

597 **A131:** Thanks for the suggestions. We have done Please search “A131” for details in Section ?? of
598 the revised paper.

599

600 **Q132:** Line 165: ...; *and2*)...

601 **A132:** Thanks for the suggestions. We have done Please search “A132” for details in Section ?? of
602 the revised paper.

603

604 **Q133:** Line 263: It is easy to see ? This should not be used in a scientific paper.

605 **A133:** Thanks for the suggestions. We have done Please search “A133” for details in Section ?? of
606 the revised paper.

607

608 **Q134:** Figure 1: It is better to add frames to the subfigures A and B. To be visible on paper, the authors
609 are also advised to make a closeup on the branch to highlight the vertex distribution.

610 **A134:** Thanks for the suggestions. We have done Please search “A134” for details in Section ?? of
611 the revised paper.

612

613 **Q135:** Figure2: The authors should use the same camera or projection to render the skeleton and the
614 mesh. The morphology seems to be rendered with orthographic camera while the mesh is rendered with a
615 perspective camera, so it feels that they do not match.

616 **A135:** Thanks for the suggestions. We have done Please search “A135” for details in Section ?? of
617 the revised paper.

618

619 **Q136:** Figure 6: The closeup in red must be explained in the figure caption.

620 **A136:** Thanks for the suggestions. We have done Please search “A136” for details in Section ?? of
621 the revised paper.

622

623 **Q137:** Figures 7, 8, 9, 10, 11, 12 and 13 can be grouped all together in a single figure to demonstrate
624 the different types of operations applied on a mesh patch. They can be also moved to the supplementary
625 material with some explanation on every operation.

626 **A137:** Thanks for the suggestions. We have done Please search “A137” for details in Section ?? of
627 the revised paper.

628

629 **Q138:** The top closeup snapshots in Figure 15 should have some gaps.

630 **A138:** Thanks for the suggestions. We have done Please search “A138” for details in Section ?? of
631 the revised paper.

632

633 **Q139:** Figure 19: The closeups must have the same extent (same width, height and center to be able to
634 make meaningful comparisons).

635 **A139:** Thanks for the suggestions. We have done Please search “A139” for details in Section ?? of
636 the revised paper.

637

638 **Q140:** The captions of all the figures must be revisited. They must be informative and descriptive, not
639 just titles.

640 **A140:** Thanks for the suggestions. We have done Please search “A140” for details in Section ?? of
641 the revised paper.

642

643 **Q141:** The captions of all the figures must be revisited. They must be informative and descriptive, not
644 just titles.

645 **A141:** Thanks for the suggestions. We have done Please search “A141” for details in Section ?? of
646 the revised paper.

647

648 **A.2 Reviewer #2:**

649 The manuscript deals with surface meshing of a neuron’s plasma membrane. The methodology uses
650 graph information of the cell and vertex-based diameter information to construct 2d manifolds represented
651 by a triangular surface grid. The key idea is to start with a sphere (the soma) and deform the mesh along
652 the graph given by the input data. The authors point out that this approach is novel in the landscape of
653 other tools that deal with the same problem (i.e. generating surface meshes from point-diameter input).
654 Since there are numerous methods available in the literature, the authors rightly compare their method to
655 existing methods, with the intent of demonstrating that the presented method is more reliable and will
656 produce better meshes than existing tools. This is where major concerns about the current manuscript
657 arise.

658 **Q201:** The authors state that the presented method should produce high-quality meshes, guaranteeing
659 “topological robustness”. Topological robustness seems to be defined by a surface mesh that contains
660 no holes and contains regularized triangles. The motivation for this is to provide meshes suitable for
661 3D modeling and simulation of neuronal processes. However, the manuscript lacks any evidence that
662 the resulting meshes are indeed suitable for numerical simulation. It is unclear how many degrees of
663 freedom meshes will typically contain. A perfectly regularized triangular grid may provide a great visual
664 representation of a neuron, but a tetrahedral volume mesh generated from the surface mesh might be too
665 large to even fit on a single node of a high-performance computer. Also, there is no consideration of the
666 smoothness of the vector field defined by the normal vector of each triangle. Without this it may be very
667 difficult to iteratively coarsen the mesh. If the focus is to provide meshes for numerical simulation, it
668 would seem more appropriate to use anisotropic elements from which, e.g., hexahedral volume meshes
669 could be constructed. Anisotropic elements would be advantageous in discretizing neurites.

670 **A201:** Thanks for the suggestions. We have done Please search “A201” for details in Section ?? of
671 the revised paper.

672

673 **Q202:** The authors don’t provide any evidence that the meshes can be used in numerical simulations
674 (which, see above point, is the main driver of this work).

675 **A202:** Thanks for the suggestions. We have done Please search “A202” for details in Section ?? of
676 the revised paper.

677

678 **Q203:** The meshing procedure is said to be robust in terms of variable input data. However, only a
679 single example is used to compare the authors’ method with existing methods. It is unclear how the
680 method works on a large set of neuron data. If the authors can demonstrate that the method produces
681 high-quality meshes for a random sample set of e.g., 1,000 neurons from NeuroMorpho.org, this claim
682 could be partially substantiated. But from a single neuron example it remains unclear if the method is truly
683 superior to existing methods.

684 **A203:** Thanks for the suggestions. We have done Please search “A203” for details in Section ?? of
685 the revised paper.

686

687 **Q204:** The authors make no mention of runtimes. How long does it take to produce a computational
688 mesh of a single neuron?

689 **A204:** Thanks for the suggestions. We have done Please search “A204” for details in Section ?? of
690 the revised paper.

691

692 **Q205:** It would be good for reviewers to test the method. Is the code accessible somewhere?

693 **A205:** Thanks for the suggestions. We have done Please search “A205” for details in Section ?? of
694 the revised paper.

695

696 **Q206:** The actual method is hardly described. There is mention of an OpenMesh library, but it is
697 completely unclear what this library does and provides. The query strategy is not described in a way that
698 can be followed. For instance, the authors talk about probing which vertices are close to a current node.
699 However, no distance measure is provided. What does close mean in this context? If the euclidian distance
700 is used, then what happens if two neurites come very close to one another? Then vertices may be identified
701 as close, although they are far away from one another on the graph. Figure 4 provides nearly no additional
702 insight, the figure legend is two words. It appears that section 2.4 contains the new intellectual portion of
703 the work. Again, it remains extremely nebulous as to what is actually done in this part of the algorithm,
704 nor is there an informative outline of the algorithmic steps. Equation (1) uses a skeletonV, which is not
705 defined. The quartic kernel function in eq. (3) is said to be a good choice, again no reasoning is provided.
706 It is not clear how eq. (4) is connected to eq. (2), i.e., what is $F_i(p)$ vs. $F(p)$? And why is $F_{Quartic}(p)$ and
707 T in eq. (5) independent of subscript i? The same question holds for eq. (6). Given that section 2.4 is the
708 only novel part there is no way to judge the quality of the approach.

709 **A206:** Thanks for the suggestions. We have done Please search “A206” for details in Section ?? of
710 the revised paper.

711

712 **Q207:** The remeshing section 2.5 is more descriptive than 2.4 which is surprising, since the methods
713 presented in 2.5 seem to be standard remeshing strategies, consisting of edge collapse, split, and flip
714 operations. If there is new intellectual merit here, the authors need to clearly point out differences to
715 existing methods.

716 **A207:** Thanks for the suggestions. We have done Please search “A207” for details in Section ?? of
717 the revised paper.

718

719 **Q208:** Section 2.6 discusses adaptive meshing. The authors rightly point out that a higher mesh resolution
720 is needed in regions with high curvature. However, the authors then use branch point location as the only
721 identifier for high-resolution regions of the cell. I can't see how this would identify e.g. neurites that take
722 sharp turns (close to 180 degree angles). I agree that adaptive remeshing is needed, but I am very doubtful
723 that neglecting discrete curvature and only relying on branch point locations will work.

724 **A208:** Thanks for the suggestions. We have done Please search “A208” for details in Section ?? of
725 the revised paper.

726

727 **Q209:** In the introduction the authors say “there is a growing requirement to simulate cellular behavior
728 in a high-fidelity dimension” (line 27), but no examples, references etc. are provided to support this claim.

729 **A209:** Thanks for the suggestions. We have done Please search “A209” for details in Section ?? of
730 the revised paper.

731

732 **Q210:** In sec. 2.2 (line 90): “the notion of short neurite segments are to be formalized using violation
733 categories (cf. Figure 3), but I don't see anything in Fig. 3 that highlights any violation categories.

734 **A210:** Thanks for the suggestions. We have done Please search “A210” for details in Section ?? of
735 the revised paper.

736

737 **Q211:** Section 2.5, lines 165-166, the statement about “allowing numerically stable calculations” is
738 entirely unsupported. It remains unclear whether the produced meshes can be used in numerical simulations.
739 Also, the notion of stability is not defined.

740 **A211:** Thanks for the suggestions. We have done Please search “A211” for details in Section ?? of
741 the revised paper.

742

743 **Q212:** The remeshing algorithm provides no real information for the reader.

744 **A212:** Thanks for the suggestions. We have done Please search “A212” for details in Section ?? of
745 the revised paper.

746

747 **Q213:** Figure 8 is unclear to me.

748 **A213:** Thanks for the suggestions. We have done Please search “A213” for details in Section ?? of
749 the revised paper.

750

751 **Q214:** Line 277-278: This statement is unsupported. Do stitching operations typically introduce cracks
752 or non-smoothness? No examples or literature is provided.

753 **A214:** Thanks for the suggestions. We have done Please search “A214” for details in Section ?? of
754 the revised paper.

755

756 **Q214:** Line 277-278: This statement is unsupported. Do stitching operations typically introduce cracks
757 or non-smoothness? No examples or literature is provided.

758 **A214:** Thanks for the suggestions. We have done Please search “A214” for details in Section ?? of
759 the revised paper.

760

761 **Q215:** Review of CTNG (starting line 292). While it may be true that the marching cubes algorithm will
762 create triangles with greatly varying aspect ratios, wouldn’t a post-processing step as described in sec. 2.5
763 suffice?

764 **A215:** Thanks for the suggestions. We have done Please search “A215” for details in Section ?? of
765 the revised paper.

766

767 **Q216:** All figure captions are uninformative, and for some figures I frankly can’t understand their
768 meaning.

769 **A216:** Thanks for the suggestions. We have done Please search “A216” for details in Section ?? of
770 the revised paper.

771

772 **Q217:** There are many typos and grammatical issues.

773 **A217:** Thanks for the suggestions. We have done Please search “A217” for details in Section ?? of
774 the revised paper.

775

776 **A.3 Reviewer #3:**

777 This paper proposed a neural morphology modeling based on skeleton-driven progressive adaptive
778 remeshing for 3D neuron surface. The model achieves both smooth surface approximation and high-
779 quality mesh vertex distributions at the neuron branches. Here are my comments and questions that should
780 be addressed by the authors:

781 **Q301:** Due to the microscopy resolution limitations, it is difficult to reconstruct the 3D surface of
782 neuron branch accurately for even professional annotators. Thus, most researchers use point-and-diameter
783 approximation to focus on tracing the centerline of the neuron branch, which can represent the neuron
784 morphology. The author should give more detailed advantages or application examples of surface meshing
785 method to prove the significance of the surface reconstruction.

786 **A301:** Thanks for the suggestions. We have done Please search “A301” for details in Section ?? of
787 the revised paper.

788

789 **Q302:** The style of Fig. 3 in this paper is similar with Fig.2 in Ref. [1]. Thus, Fig. 3 should be redrawn.
790 [1] Mörschel K, Breit M, Queisser G. Generating neuron geometries for detailed three-dimensional
791 simulations using anamorph[J]. Neuroinformatics, 2017, 15(3): 247-269.

792 **A302:** Thanks for the suggestions. We have done Please search “A302” for details in Section ?? of
793 the revised paper.

794

795 **Q303:** For Fig.16, the sub-images need to be aligned.

796 **A303:** Thanks for the suggestions. We have done Please search “A303” for details in Section ?? of
797 the revised paper.

798

799 **Q304:** In Section 3.2, the author did not give any qualitative or quantitative analysis of the proposed
800 model results. The audience will be confused about the effect of the proposed model. What’s more, please
801 show the original images to illustrate neuron branch characteristics.

802 **A304:** Thanks for the suggestions. We have done Please search “A304” for details in Section ?? of
803 the revised paper.

804