

Scalable Blocked Volume Meshing of Large Vascular Networks

ANONYMOUS AUTHOR(S)
SUBMISSION ID: PAPER 1528

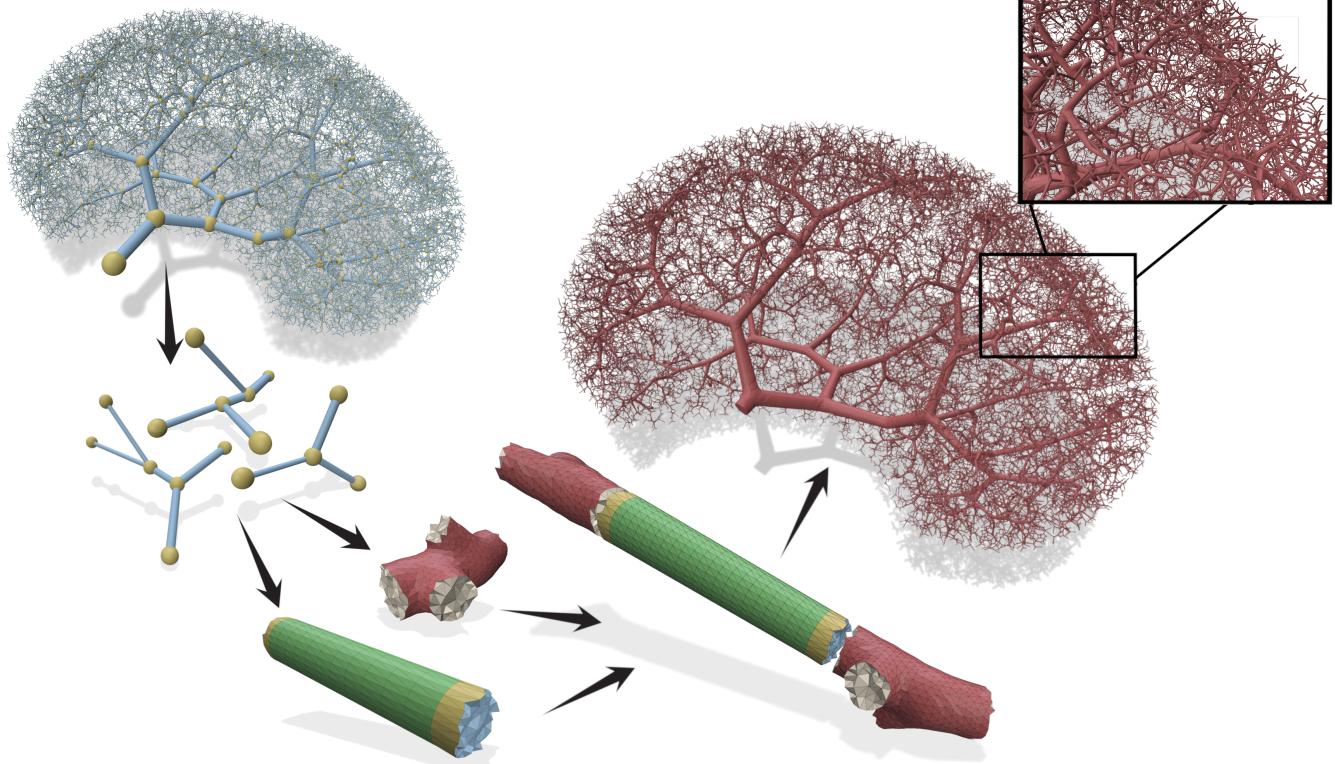


Fig. 1. The blocked mesh computation process for a renal vascular network. The skeleton structure is divided into isolated clusters, which are meshed independently and joined back together by tubular mesh pieces to produce the full mesh. The blocked approach allows using variable resolution and parallel computation for performance scalability.

Understanding blood flow in renal arterial networks is a critical area of clinical research. However, computational fluid dynamics (CFD) simulations are hindered by the absence of a full-scale volume mesh model. Recent advancements have enabled the generative reconstruction of full-scale vascular structures in a rat kidney, producing a skeleton tree structure with corresponding vessel radii.

Conventional skeleton-based mesh generation techniques are impractically slow due to the wide variability in vessel scales. To address this, we propose a novel multi-scale inverse skeletonization method that employs a blocked approach, allowing local scales to guide the mesh generation process. This method involves dividing the skeleton into clusters by cutting long

vessel edges, followed by independent meshing of each cluster. In a second phase, the cluster meshes are seamlessly connected using tubular segments.

A key requirement of this approach is the presence of sufficient "cuttable" locations in the skeleton structure, a condition not always met by generative reconstructions. To overcome this, a preprocessing step is introduced to enhance the geometric and anatomical plausibility of the tree embedding.

Our proposed method successfully generates anatomically realistic volume mesh models of the vascular network. Furthermore, the algorithm is embarrassingly parallel, demonstrating excellent scalability with increasing network size.

CCS Concepts: • Computing methodologies → Physical simulation; Mesh models; Volumetric models; Mesh geometry models.

Additional Key Words and Phrases: Vascular Networks, Volume Meshing, Tetrahedral Meshing, Large scale meshes

ACM Reference Format:

Anonymous Author(s). 2025. Scalable Blocked Volume Meshing of Large Vascular Networks. In *Proceedings of ACM (SIGGRAPH)*. ACM, New York, NY, USA, 10 pages. <https://doi.org/XXXXXXX.XXXXXXX>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGGRAPH, Vancouver

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2025/08
<https://doi.org/XXXXXXX.XXXXXXX>

115 1 Introduction

116 The kidney, with its unique vascular structure, forms a well-balanced
 117 distribution network whose adaptation properties cannot be as-
 118 sessed yet through modern experimental techniques despite its
 119 importance for learning about drug delivery and lifestyle diseases.
 120 Computational Fluid Dynamics (CFD) simulation is the only viable
 121 approach for this line of research as it is non-invasive and allows
 122 confirmation of first principles and models. Unfortunately, the sim-
 123 ulation approach is currently blocked by the fact that no one has a
 124 computational volume mesh of a full-scale renal vascular network.
 125 This is the problem addressed by the present paper.

126 Using a traditional segmentation pipeline [8, 20, 34] to construct a
 127 mesh is infeasible due to a number of reasons; First, current scanning
 128 technology can only resolve either the large branches in the renal
 129 arterial network through low resolution imaging modalities [11, 31],
 130 or a small patch of the kidney in higher resolution [18, 24]. A high
 131 resolution scan of the entire structure is not possible, and would
 132 entail lethal radiation doses. Second, even in the presence of such an
 133 image, traditional segmentation techniques would produce meshes
 134 with equivalently high mesh resolution, making them too large
 135 for simulation processing. Fundamentally, the large-scale variance
 136 of the vessel structures in kidneys makes any single-resolution
 137 approach bound to fail.

138 Recent research has enabled generative reconstruction of skeleton
 139 representations of the full-scale vessel structures of rat kidneys [33].
 140 Here, the vessel network is represented as a graph, with edges mod-
 141 eling tubular vessels, and nodes (with accompanying radius data)
 142 modeling branch points. Already this line of research has shown
 143 its potential for synthetic data generation in other types of tissue
 144 than rat kidneys [35]. However for blood flow simulation, the tech-
 145 nique is still limited since the skeleton models are not compatible
 146 with CFD. It is therefore necessary to convert the skeleton model to
 147 a computational mesh ('inverse skeletonization'). Although many
 148 techniques for this exist, they all have either resolution issues like
 149 the traditional pipeline, or result in non-anatomical geometries (see
 150 Section 2). Thus, an anatomically plausible, scale-adaptable inverse
 151 skeletonization technique is needed.

152 We propose a divide and conquer style solution, based on dividing
 153 the skeleton into easily re-connectable parts, and computing volume
 154 meshes for each part independently and in parallel. We propose a
 155 data refinement step to improve the geometry of the skeleton and
 156 establish stronger geometrical invariants for the algorithm to rely
 157 on.

158 The overall benefit of our novel contributions is to provide an
 159 algorithmic framework that

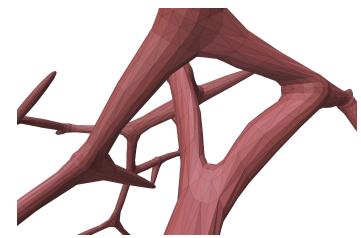
- 160 • Enables flexible refinement of generative skeleton models
 161 to ensure geometrical properties
- 162 • Exploits a blocked parallel approach to achieve an efficient
 163 two-pass meshing of large vessel structures.
- 164 • Applies mesh generation techniques known for organic-
 165 looking tubular mesh geometry.
- 166 • Is focused on generation of volumetric meshes suitable for
 167 simulation and not visualization.
- 168 • Has code and data available at: [Published upon acceptance]

169 2 Related Work

170 Skeleton models have long been useful as an abstraction of complex
 171 geometry, and in some instances as a computational model [7, 25],
 172 but in many cases it is necessary to convert the skeleton to a mesh.
 173 As such, many techniques for inverse skeletonization exists.

174 Some techniques directly construct a rough mesh from the skele-
 175 ton by various algorithms and heuristics. Examples include B-meshes
 176 [14], scaffolding [23], SQM [2], FEQ [22], that all use "key-balls",
 177 which are user defined spheres centered at skeleton nodes, as a
 178 guide for the mesh generation. These techniques are often com-
 179 bined with mesh subdivision algorithms [6] to improve and smooth
 180 the produced geometry, and they tend to focus on creating quad-
 181 dominant surface meshes due to their nice edge flow properties
 182 desired in sculpting. For CFD simulations, those considerations are
 183 irrelevant, and as all these techniques output surface meshes rather
 184 than volume meshes, these will at most be useful as an in-between
 185 step for generating a high-quality tetrahedral volume mesh.

186 Furthermore, the geomet-
 187 tries of these techniques are
 188 usually not anatomically plau-
 189 sible. B-meshes are inherently
 190 rectangular, and even after
 191 smoothing, the geometry is
 192 not always radially symmet-
 193 ric. FEQ has trouble with very
 194 long edge segments, and in-
 195 serting intermediate vertices
 196 on each segment results in
 197 warped branch geometries,
 198 as seen on Figure 2.



199 Fig. 2. Vessel structure surfaces gen-
 200 erated by FEQ. Geometry is not anato-
 201 mically ideal, with bulging vessels and
 202 warped branch geometries.

203 An alternative approach, which is often preferred in medical con-
 204 texts due to its more anatomically realistic shapes, is iso-contouring.
 205 First described by Bloomenthal [4, 5], this is a broad mathematical
 206 framework where the level-sets of an implicit function defines the
 207 surface, which is computed using contouring techniques such as
 208 marching cubes [16, 26]. The choice of implicit function is critical,
 209 with Gaussian convolution filters along the skeleton, as described
 210 in Oeltze and Preim [21], usually being preferred as they have been
 211 experimentally proven as better compared to eg. meta-balls [3], dis-
 212 tance fields [32], winding numbers [13], and neural signed distance
 213 fields [17].

214 These techniques require a global sampling resolution to be cho-
 215 sen for the implicit function, and the contoured meshes are only
 216 accurate when the resolution is high enough to capture all details in
 217 the implicit function. Thus the resolution is dictated by the size of
 218 the smallest structure, which for our application results in infeasibly
 219 slow performance and needlessly detailed meshes.

220 The methods described above all produce surface meshes, which
 221 are useful for rendering as well as some types of surface-based sim-
 222 ulation, but not for CFD simulation. Nevertheless, they can function
 223 as a stepping stone towards volume meshes, by serving as input to
 224 an automatic tetrahedralization algorithm. Many such algorithms
 225 exist with different requirements and capabilities [1, 19, 27]. TetGen
 226 [28] is a common framework for automatic tetrahedralization. It

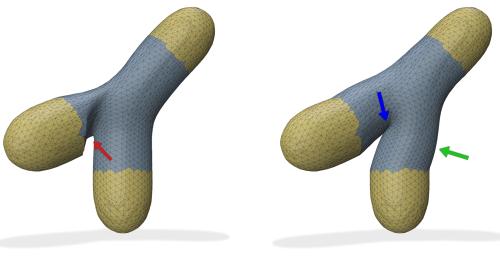


Fig. 3. Deformed vessel branches. Bi-harmonic deformations (left) does not preserve tubular shapes (red arrow), and As-Rigid-As-Possible deformations (right) produce bent vessels (green arrow) and creases (blue arrow).

requires a manifold surface (closed and without self-intersections), and is able to leave the input surface as is. Another alternative is TetWild [9], which can process arbitrary triangle soups, but does not preserve the input geometry.

One commonality for tetrahedralization algorithms however, is that while it is usually possible to specify a desired cell size, that size is applied globally. Thus, for multi-scale structures like vascular networks, these techniques have the same time and size complexity issues as iso-surface contouring.

A third approach is to attempt to skip the surface mesh entirely and compute a volume mesh directly from the skeleton. One method is to utilize deformation fields to deform known prototype volume meshes to match the desired vessel geometry. By considering each vessel branch in the tree as a separate part, prototype branch meshes could be deformed to make vessels that have the correct direction and radius while maintaining easily connectable, known surface interfaces. Two common deformation techniques are Bi-harmonic deformations [12] and 'As Rigid As Possible' deformations [29], both of which are computed from manually placed key-handles.

Unfortunately, as seen on Figure 3, both of these approaches are prone to producing creases in the geometry when the deformations become significant, and the geometries are not consistently anatomically realistic, exhibiting bulges and concave artifacts. Figure 12 shows a parameter study of deformation techniques.

When working with skeleton and mesh data structures, the structure sizes often make smoothing techniques the only viable approach for modification. Common smoothing types include Laplacian and Taubin smoothing [30], but many complex schemes exist as well. Further, although not originally proposed as a deformation technique, recent research [10] has applied barrier potentials as a convenient way to quantify self-intersections in geometric models.

3 Scalable Vascular Volume Meshing

Clearly, what is needed is a scale-adaptable approach which utilizes the regular and hierarchical structure of the vascular network to avoid global computation complexities. Our approach is a blocked algorithm, where volume meshes for each complex part of the vascular structure is generated separately allowing appropriate sample and mesh resolutions to be used. Due to their advantage in generating anatomically plausible geometries, we use convolution surfaces combined with TetGen tetrahedralization to generate the volume

meshes. The complexity of the algorithm then lies in how to divide the skeleton and how to reconnect the meshes.

We exploit the regular geometric structure of vessel networks, which are modeled by the skeleton representation as straight tubes joining together at branch nodes. In particular, as long as two branches are sufficiently separated, the geometry of the vessel between them is a simple tube whose radius interpolates between the two branch node radii. Branches which are not sufficiently separated exhibit complex inter-dependent geometries, which are not easily solved independently, thus such branches will be meshed together as a *cluster*. By meshing the clusters independently (cutting only simple edge between them), the merging process is reduced to connecting tubular mesh pieces.

The details of these steps are discussed below, however it is necessary to first consider that this approach only works when the skeleton has sufficient quantities of *cuttable* vessels. If there are too few, the clusters will become large and include vessels of significantly different size and the ability to choose an appropriate local resolution is lost. There are 3 conditions that are required for a vessel to be *cuttable*:

- (1) **Intersection:** The vessel must not intersect any other part of the cluster, as the geometries and meshes of the intersecting pieces then merge.
- (2) **Length:** The vessel must be long enough that the mesh between the two branches becomes essentially a tube.
- (3) **Angle:** The angle between the vessel to be cut and other outgoing vessels of the branch cannot be too small, as the vessel geometries then also overlap and merge.

3.1 Skeleton Data Refinement

The internal structure and geometry of the reconstructed skeleton, produced by [33] is statistically determined to match certain morphological properties, and is therefore not guaranteed to be geometrically and anatomically sound. In particular, the skeleton contains many structures that would intersect each other when transformed into vessel meshes.

Given the generative nature of the data, we propose using a set of smoothing operations on the dataset prior to mesh generation to clean up the data and improve the geometry with respect to the *cuttability* conditions. The smoothing operations will be applied iteratively, controlled by learning rate parameters and with capped modification sizes each iteration to avoid over-corrections.

Before smoothing, the skeleton subdivided by splitting nodes with valency greater than 3 until all nodes have valency of 3 or less. This is to allow maximal freedom in the structure of the network; the geometry of a high valency node is modeled just as well by multiple low-valency nodes in close proximity, but the latter allows for more flexibility.

3.1.1 Intersection - Barrier optimization. We propose a novel application of barrier potentials [10] for skeleton improvement. Barrier potentials are usually employed in collision detection, and are defined in terms of a barrier kernel, $\mathcal{K}(d, h)$, with the properties

$$\mathcal{K}(d, h) \rightarrow 0 \quad \text{for } d \rightarrow h, \quad (1)$$

$$\mathcal{K}(d, h) \rightarrow \infty \quad \text{for } d \rightarrow 0. \quad (2)$$

where d is intended as a distance measure and h is a 'support radius'. By letting d be a measure of the distance between edges, and h be the smallest permissible distance, positive values of $\mathcal{K}(d, h)$ encodes self-intersections in the skeleton, with gradients that are suitable for minimization.

As there are no unique distance and radius measures that fully captures the relationship of two 3D edges, we define the barrier potential as a double integral over the spans of pairs of edges. Let e be an edge in the skeleton and $\vec{p}_e(t)$ be a position function spanning the edge for $t \in [0, 1]$. Then the distance between points on edges e and g is

$$d_{e,g}(s, t) \equiv \|\vec{p}_e(s) - \vec{p}_g(t)\| \quad (3)$$

for $s, t \in [0, 1]$. As support radius we use a multiple of the sum of the edge radii, however as the radii of edges vary across their lengths, these also need range parameters. Thus letting $r_e(t)$ be a function spanning the radius range of edge e for $t \in [0, 1]$, the proper sum of radii of edges e and g is

$$h_{e,g}(s, t) \equiv r_e(s) + r_g(t) \quad (4)$$

The barrier potential for any pair of edges e and g in the skeleton is then given by a double integral over their ranges

$$\psi_{e,g} \equiv \int_{t=0}^1 \int_{s=0}^1 \mathcal{K}(d_{e,g}(s, t), h_{e,g}(s, t)) ds dt, \quad (5)$$

which is a value that is positive whenever the two edges intersect, and tends towards infinity as the overlap increases. For the entire skeleton, the potential is simply the sum of the relevant edge-pair potentials

$$\psi \equiv \sum_{e \in \mathcal{E}} \sum_{g \in \mathcal{E} \setminus \mathcal{N}_e} \psi_{e,g} \quad (6)$$

where \mathcal{N}_e is the 'neighborhood' of edge e , that is, set of edges that are connected to one of the endpoints of e (including itself). In practice, it is useful to improve the computational efficiency by only computing the potential for edge pairs which share an axis aligned bounding box overlap, and these relationships can be precomputed and recomputed in regular intervals, provided the vertex movements are small.

Minimization with respect to the barrier potential will seek to resolve self-intersections, thus by implementing a quadrature estimation of the integral sum in an autograd framework, the vertices can be smoothed by stepping in the direction opposite the gradient

$$\vec{v} \leftarrow \vec{v} - \ell_1 \nabla_{\vec{v}} \psi \quad (7)$$

where ℓ_1 is rate parameter.

3.1.2 Length - Repel smoothing. To bias the skeleton structure towards cuttable edges, a simple repelling force is added to the smoothing scheme. The barrier kernel, \mathcal{K} , from above is reused to produce weights based on the distances between nodes. In particular, each node v is repelled by its neighbor w proportionally to the value of the barrier kernel evaluated with the distance between nodes and using radius of the node as support. Summing over all neighbors, $w \in \mathcal{N}_v$, the step is

$$\vec{v} \leftarrow \vec{v} - \ell_2 \sum_{w \in \mathcal{N}_v} \mathcal{K}(\|\vec{v} - \vec{w}\|, \gamma r_v) \vec{u}_{w,v} \quad (8)$$

where ℓ_2 is another rate parameter and γ is a constant factor, chosen in our implementation to be 4, r_v is the radius of vertex v and $\vec{u}_{w,v}$ is a unit vector in the direction from w to v .

3.1.3 Angle - Weighted Laplacian style smoothing. Finally, to improve the angular distribution of vessels around the nodes, we propose a Laplacian style scheme. Note, that simple Laplacian smoothing, where vertices are moved towards the mean position of its neighbors, does not necessarily balance the angles.

Instead, we propose a scheme where each pair of neighbors is considered, and a force which is inversely proportional to the angle spanned by their vessels is applied towards their mean position. Specifically, for each pair of neighbors w and w' of vertex v , we define a weight as

$$\eta_{w,w'} \equiv \frac{1}{\min(\phi_{w,w'}, \lambda)} \quad (9)$$

where $\phi_{w,w'}$ the angle between the vessels towards w and w' and λ is some lower bound, in our implementation chosen to be 0.1. Further, letting $\vec{m}_{w,w'}$ be the displacement vector from vertex v to the midpoint between w and w' , the smoothing force is then defined as a weighted average of the displacements

$$\vec{v} \leftarrow \vec{v} - \ell_3 \frac{1}{\eta_T} \sum_{(w,w') \in \binom{\mathcal{N}_v}{2}} \eta_{w,w'} \vec{m}_{w,w'} \quad (10)$$

where ℓ_3 is yet another rate parameter, η_T is the sum of the weights and $\binom{\mathcal{N}_v}{2}$ is the set of all unordered pairs of elements of \mathcal{N}_v .

For leaf and root nodes in the tree, Laplacian style smoothing risks shrinking the geometry, however since these nodes only has one neighbor, the definition is void and no changes are thus applied to such nodes.

3.2 Blocked Volumetric Mesh Generation

3.2.1 Cluster partitioning. The first step in the algorithm is to partition the skeleton tree into clusters. Ideally, the tree is partitioned into as small clusters as possible, while maintaining easily re-connectable cuts. To achieve this, we derive the geometric conditions that are necessary for the edge to be *cuttable*. It is not practical to check for self-intersections at runtime, thus we assume this issue has been alleviated by the refinement process, and focus on the **length** and **angle** conditions.

Let ω be some margin factor which is multiplied on the conditions to ensure some leeway, chosen in our implementation to be 1.1, and r_1, r_2 be the radii of the nodes connected by an edge e . The first *cuttability* condition ensures that the **length** of the edge is sufficient, by checking that the edge is longer than twice the sum of the radii;

$$|e| > 2\omega(r_1 + r_2) \quad (11)$$

The second *cuttability* condition ensures the edge is not overlapping with a neighbor edge. This becomes an issue when the angle between edges are small, and the their lengths are not sufficiently long to separate them. In particular, the vessels will not have space to properly separate unless

$$|e| > \frac{r_{e'}}{\sin(\phi)} + \frac{r_e}{\tan(\phi)} \quad (12)$$

457 holds, where r_e and $r_{e'}$ are the average radii of edge e and a neighbor
 458 edge e' , respectively, and ϕ is the angle between the edges. See the
 459 supplementary material for a derivation of the expression.

460 An edge is cuttable whenever both conditions are satisfied, and
 461 the condition expressions are used to determine the cut distance.
 462 Once an edge has been cut, the node it connected to can be analyzed
 463 recursively to create another cluster. If an edge could not be cut, the
 464 node is added to the current cluster and its edges are examined.

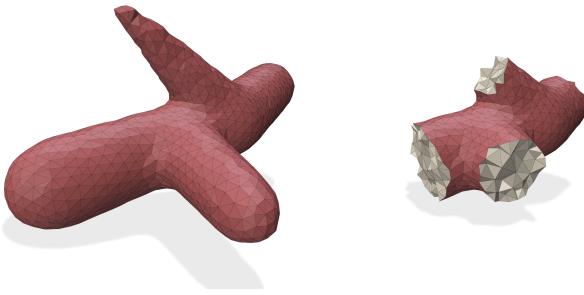
465 **3.2.2 Cluster mesh generation.** After the skeleton has been parti-
 466 tioned, the first pass of the meshing algorithm runs on each cluster
 467 in parallel. A graph representing the cluster section of the tree is
 468 created and a convolution over those edges with a Gaussian kernel,
 469 \mathcal{G} , is computed. Specifically, for each point p in a grid with some
 470 resolution we compute the function

$$\phi(\vec{p}) \equiv \int_S \mathcal{G}(|\vec{p}' - \vec{p}|) d\vec{p}' \quad (13)$$

471 where S is the set of points in the cluster graph. For details about
 472 efficient computation of convolution surfaces, see [4].

473 An off-the-shelf marching cubes implementation is then used to
 474 convert the implicit field to a surface mesh, and Taubin smoothing
 475 [30] is applied to improve the regularity of the mesh structure, as this
 476 is important to make the tetrahedralization algorithms produce high
 477 quality volume meshes. Next, the surface is run through TetGen,
 478 producing a volume mesh of the cluster. Note, that the resolution of
 479 both the surface and volume mesh is indirectly controlled by the
 480 resolution of the implicit convolution field, and we can tune that to
 481 achieve a resolution that is suitable for the specific cluster.

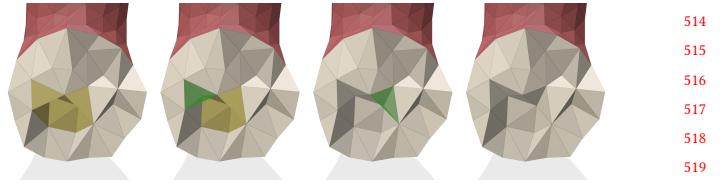
482 Figure 4 (left) shows the result of one of these mesh genera-
 483 tions for a cluster with 2 nodes and 4 outgoing vessels. As seen, the
 484 complex branching geometry has successfully been constructed,
 485 but the outgoing vessels are generated with round tips. Before the
 486 cluster meshes can be connected to construct a combined mesh,
 487 these must be trimmed and cleaned.



503 Fig. 4. Generated cluster volume mesh before and after trimming.

504 The cut distances computed during the cluster analysis, together
 505 with the directions of the vessels define planes by which the mesh
 506 is trimmed. Each trimming is achieved by identifying which
 507 vertices lie beyond the plane, and removing tetrahedra which have all
 508 four vertices beyond the plane. This exposes a number of internal
 509 triangles, which forms the *end surface*, as shown on Figure 4 (right).

510 The second step of the algorithm needs to compute volume
 511 meshes for a connector tube which joins two such *end surfaces*
 512 together. This can in general be prone to instabilities and edge cases,



514
 515
 516
 517
 518
 519
 520
Fig. 5. Flattening algorithm, identifying triangle overlaps (yellow) and re-
moving attached tetrahedra (green) until the *end surface* is *projection flat*.

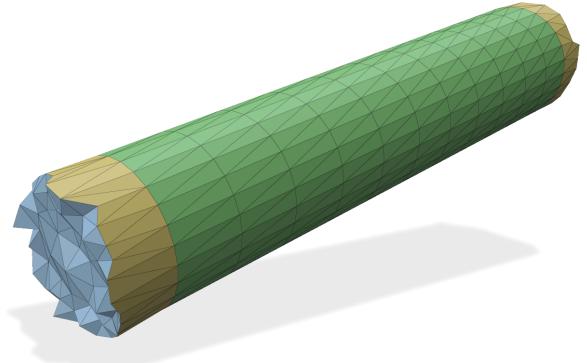
521
 522
 523
 524
 525
 526
 527
 528
 529
 530
 531
 532
 533
 534
 535
 536
 537
 538
 539
 540
 541
 542
 543
 544
 545
 546
 547
 548
 549
 550
 551
 552
 553
 554
 555
 556
 557
 558
 559
 560
 561
 562
 563
 564
 565
 566
 567
 568
 569
 570

as the *end surfaces* are not necessarily flat. Thus to make the following step simpler, we wish to have some guarantees about the structure of the *end surface*. In particular, we want the end surface to be *projection flat*; When projected onto the cut plane, there should not be any overlap.

This property is not guaranteed when simply pruning tetrahedra as described, and we therefore propose the following simple greedy algorithm to refine the *end surface*; while the surface is not *projection flat*, select one of the triangles in the overlap and prune the tetrahedron it is part of. Add the newly exposed triangles to the *end surface*. Repeat until the surface is *projection flat*. While we do not prove that this algorithm eventually finds a *projection flat end surface*, in practice it terminates in only a few steps. Figure 5 shows the process.

3.2.3 Vessel generation. We now have volume meshes for each cluster, terminating in well-formed *end surfaces*. To construct the complete structure, a method for merging the volume meshes is needed.

To avoid the complexity of manually constructing high quality volume meshes, we propose an algorithm that again defers this task to TetGen. To achieve this, a tubular surface mesh, with ends that perfectly matches the *end surfaces* of the connected clusters is generated, and then given to TetGen, which is configured to leave the input surface in place. Configuring TetGen like this means that the quality of the volume mesh is highly dependent on the quality



567
 568
 569
 570
Fig. 6. Constructed vessel surface to connect cluster meshes. The *end surfaces* of the connected clusters are copied (blue) and merged with a regular cylindrical mesh (green) by a triangle strip (yellow).

571 of the surface mesh, which is why we have taken steps to ensure
 572 the *end surfaces* are relatively simple.

573 The tubular surface is modeled by a regular cylindrical quad mesh
 574 section with radius ranging to match the geometry of the true vessel.
 575 The radial resolution of the tube is chosen to be as close as possible
 576 to the number of vertices on the edges of the two *end surfaces*, and
 577 the length of the cylinder is chosen to be 10% shorter than the
 578 distance between the *end surfaces*. This leaves a small gap between
 579 the cylinder and each of the *end surfaces*, in which we weave a
 580 triangle strip, connecting nodes by processing them in order by
 581 radial angle.

582 Figure 6 shows a vessel surface constructed by this algorithm. Due
 583 to the flatness property of the *end surfaces*, this mesh is manifold,
 584 and TetGen can therefore automatically compute a volume mesh
 585 with this surface as its boundary. Since the surface is chosen with
 586 vertices that match the *end surface* of each cluster mesh, we can
 587 easily merge the three by combining equivalent vertices.

588 Repeating this throughout the tree, the full mesh is constructed.

589
 590 **3.2.4 Mesh smoothing.** While the method as described above works
 591 and generates valid volume meshes, the geometric embedding of
 592 the tetrahedra around the merged areas can be irregular. For CFD
 593 simulation stability, it is important that the mesh does not have
 594 sudden changes in internal geometry, and we therefore suggest
 595 smoothing the cells around the merged area slightly, to improve
 596 their shapes.

597 Simple Laplacian smoothing is common for volume meshes, but
 598 has the unintended effect of shrinking the geometry. We therefore
 599 instead propose a 2-step smoothing approach, in which we
 600 first smooth the surface of the mesh around the merged area using
 601 Taubin smoothing (which keeps sizes steady), and afterwards use
 602 the surface as a fixed boundary condition for Laplacian smoothing
 603 of the internal cells. This ensures both parts of the mesh is smoothed
 604 while keeping the geometric shape of the mesh fixed.

606 4 Results

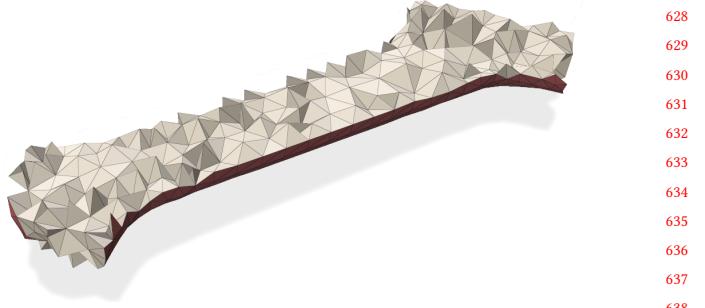
607 Our implementation is written in Python and the source code, data
 608 and results are available at [Published upon acceptance].

609 All experiments have been run on a personal laptop with an Intel
 610 Core i7-4720HQ processor, 16gb of RAM and Nvidia GeForce GTX
 611 860M graphics card.

613 4.1 Anatomical Invariants

614 The skeleton refinement algorithm is run with iterative applications
 615 of the three smoothing forces to improve the skeleton geometry.
 616 We found that rate parameters $\ell_1 = 25$, $\ell_2 = 0.01$, $\ell_3 = 0.02$
 617 resulted in balanced effects from the three functions. To examine
 618 the improvements to the geometry afforded by the modifications,
 619 we define metrics to quantify each issue;

- 620 • **Intersection:** The barrier potential, ψ , is itself an estimate
 of the amount of self-intersection present in the skeleton.
- 621 • **Length:** Quantified by the percentage of edges whose length
 is smaller than 2.5 and 3.0 times the radius of the node.
- 622 • **Angle:** Quantified by the percentage of edge pairs, whose
 internal angle is less than $\pi/10$.



623 Fig. 7. Internal element structure of a connected cluster pair. The tetrahedral
 624 elements are regular and uniform, including at the merged area.

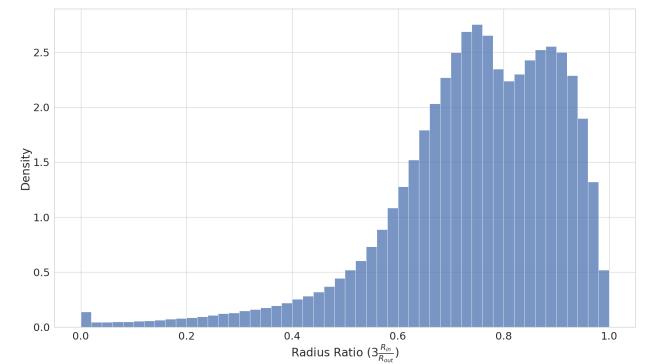
625 Figure 11 shows these metrics as well as the sizes of the vertex
 626 displacements as functions of the iteration count.

627 The method is seen to clearly reduce the prevalence of the problematic
 628 properties in the skeleton. The mean displacement of vertices is within
 629 $25\mu\text{m}$ with higher quantiles reaching overall displacements of around $200\mu\text{m}$, which, compared to the total size of
 630 the vessel structure (around $20000\mu\text{m}$), is relatively minor. While
 631 promising, the method is not able to completely free the structure
 632 of problematic vessel structures, likely because the repel-smoothing
 633 and angle-smoothing can cancel each other when the leaf nodes are
 634 fixed in place.

635 4.2 Mesh Quality

636 A central priority for the mesh generation algorithm is that the
 637 generated volume meshes have high quality elements, as they are
 638 intended for simulation use where element quality is important for
 639 stability. Figure 7 shows the internal geometry of one of the
 640 constructed vessel-branch connections, where the tetrahedra structure
 641 is seen to be uniform and regular.

642 Element quality is also assessed using one of the standard quality
 643 metrics; *radius ratio* [15], which is a *fair* metric; the values are
 644 between 0, corresponding to a degenerate cell, and 1, corresponding to
 645 an equilateral tetrahedron. Figure 8 shows a histogram of the value
 646 of this metric for the elements in the generated meshes, revealing a
 647 significant majority of high-quality elements.



648 Fig. 8. Radius ratio distribution of elements in full volume mesh. The two
 649 peaks are likely from branch and connector geometries respectively.

685 4.3 Full Vascular Network Generation

686 The primary goal of this work is to, for the first time, make anatomically
 687 plausible full size volume mesh models of the renal vascular
 688 network available. With the blocked approach of this paper, this is
 689 attainable, and examples of such meshes are shown in Figure 9.

690 The mesh size is controllable by tuning the grid size used for
 691 convolution surface contouring, and these should be proportional
 692 to the dimensions of the clusters. In our implementation, it is set to
 693 a multiple, ρ , times the smallest radius in the cluster. Using $\rho < 3$
 694 makes the process unstable, and we have therefore generated meshes
 695 with the ρ ranging from 3 to 5, resulting in mesh size between 2.5
 696 and 10 GB.

697 **4.3.1 Anatomical Assessment.** As mentioned, a focus of this work
 698 has been that the meshes retain geometric properties associated
 699 with blood vessels. While these are not definitively described in
 700 the literature, the mesh should not contain sharp bends, bulges or
 701 self-intersections, and should for the most part keep its tubular
 702 shape. By visual inspection of Figure 9, these properties are seen
 703 to be present, and compared to geometries produced by alternative
 704 techniques (Figure 2), our method is a significant improvement.

705 4.4 Computational Complexity

706 Generating the volume meshes for full vessel structures is only pos-
 707 sible due to the improved computational complexity of our blocked
 708 algorithm. In particular, the algorithm is theoretically and empiri-
 709 cally seen to scale linearly in the size of the vascular network. In
 710 contrast, traditional convolution based techniques scales by the ra-
 711 tio between the sizes of the smallest and largest structures. While
 712 the algorithm is embarrassingly parallel, our implementation is
 713 sequential for simplicity.

714 On the hardware specified above, the computation of the volume
 715 meshes takes between 2 and 15 hours, depending on the resolution
 716 multiplier. In comparison, the FEQ algorithm executes in approxi-
 717 mately 3 hours on the same structure, and a global fixed-resolution
 718 computation of convolution surfaces has not been completed, but is
 719 projected (based on smaller examples) to require up to 5 days. Fur-
 720 thermore, these alternative techniques only produce surface meshes,
 721 thus a fair comparison would require a global tetrahedralization
 722 step to be performed afterwards, which has not been attempted and
 723 is likely computationally infeasible.

727 Technique	Execution time	Mesh size
728 FEQ	0:02:45	360MB
729 Global convolution	~5:00:00	-
730 Ours ($\rho = 3$)	0:02:05	2.54GB
731 Ours ($\rho = 4$)	0:05:30	5.87GB
732 Ours ($\rho = 5$)	0:15:45	10.5GB

733 Table 1. Summary of time and space complexities of the different techniques.
 734 Note that FEQ and Global convolution are surface mesh techniques.

735 5 Discussion and Limitations

736 Automatic mesh processing and repair is difficult and error prone,
 737 and the geometries that are returned from TetGen have few regu-
 738 larity guarantees. Additionally, using generative models to produce

742 the inputs to such an algorithm gives even less control over the
 743 geometric space.

744 As such, our implementation of the algorithm is unstable in rare
 745 cases. The stability generally increases with the mesh resolution, and
 746 as discussed in section 4.3, the geometries become degenerate and
 747 irregular with $\rho < 3$. With $\rho = 3$, our implementation experiences
 748 5 failed clusters out of about 60000, or equivalently a 0.01% error
 749 rate, and with higher resolutions the error rate goes to 0. In our
 750 implementation, these errors are ignored and the failed mesh is
 751 simply not added to the result.

752 Furthermore, with the approach of the algorithm being to divide
 753 the skeleton into clusters with a consistent local scale, the method
 754 is vulnerable to input geometries where vessels of significantly
 755 different radii are close to each other and not separable. We argue
 756 that this would not happen in real vessel structures, however the
 757 generative model is not guarded against such geometries.

758 More generally, the proposed data refinement method can be seen
 759 as a gradual trade-off; The better separated the mesh is, the more
 760 likely it is that locally consistent scales exist, however the more
 761 strict does the input requirements also have to be.

762 5.1 Future Work

763 This work enables the generation of volume mesh models from
 764 skeleton representations, and has been applied to a skeleton model
 765 of the renal venous network. Naturally, for complete blood flow
 766 dynamics, the arterial network and the connections between the
 767 two are just as important.

768 We hypothesize that the data refinement methods presented in
 769 this paper can be used to generate an arterial dual of the input
 770 structure, and smooth away the geometric irregularities. Further
 771 research should also cover what computational model best captures
 772 the biological dynamics of the arterial/venous transition. Until then,
 773 this work enables CFD simulations to be run on one half of a full
 774 vascular structure.

775 6 Conclusion

776 Solving the absence of a full scale volume mesh of renal vasculature
 777 is of major importance, as it enables unprecedented research possi-
 778 bilities regarding renal blood-flow dynamics. Existing techniques
 779 are either not suited to handle the large scale variance of the net-
 780 work, giving impractical computation complexities, or not suited
 781 for the generation of anatomically feasible vessel geometries.

782 A novel, blocked approach to volume mesh construction is pro-
 783 posed, exploiting the regular and hierarchical structure of vascular
 784 networks to allow mesh construction piece-by-piece with local res-
 785 olutions.

786 This approach has been enabled by the results of recent research
 787 in vessel structure skeleton reconstruction, and an original frame-
 788 work for skeleton refinement has been designed to improve the
 789 geometry of the skeleton for algorithmic use.

790 The overall technique has been shown to successfully construct
 791 organic looking full scale volume meshes of a rat renal venous
 792 network, with high quality mesh elements and smooth blending of
 793 adaptable resolution sizes, constituting a major step forwards in the
 794 direction of in-silico renal blood flow research.

References

- [1] Pierre Alliez, David Cohen-Steiner, Mariette Yvinec, and Mathieu Desbrun. 2005. Variational tetrahedral meshing. In *ACM SIGGRAPH 2005 Courses* (Los Angeles, California) (*SIGGRAPH '05*). Association for Computing Machinery, New York, NY, USA, 10–es. doi:10.1145/1198555.1198669
- [2] Jakob Andreas Bærentzen, Marek Krzysztof Misztal, and Katarzyna Welnicka. 2012. SMI 2012: Short Converting Skeletal Structures to Quad Dominant Meshes. *Comput. Graph.* 36, 5 (aug 2012), 555–561. doi:10.1016/j.cag.2012.03.016
- [3] James F. Blinn. 1982. A Generalization of Algebraic Surface Drawing. *ACM Trans. Graph.* 1, 3 (July 1982), 235–256. doi:10.1145/357306.357310
- [4] Jules Bloomenthal. 1996. *Skeletal design of natural forms*. Ph.D. Dissertation. CAN. UMI Order No. GAXNN-03088.
- [5] Jules Bloomenthal. 1997. Bulge Elimination in Convolution Surfaces. *Computer Graphics Forum* 16 (1997). <https://api.semanticscholar.org/CorpusID:37607904>
- [6] E. Catmull and J. Clark. 1978. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design* 10, 6 (1978), 350–355. doi:10.1016/0010-4485(78)90110-0
- [7] Bradley Feiger, Ajay Kochar, John Gounaris, Desiree Bonadonna, Mani Daneshmand, and Amanda Randles. 2020. Determining the impacts of venoarterial extracorporeal membrane oxygenation on cerebral oxygenation using a one-dimensional blood flow simulator. *J Biomech* 104 (March 2020), 109707.
- [8] Torkan Gholamalizadeh, Faezeh Moshfeghifar, Zachary Ferguson, Teseo Schneider, Danièle Panozzo, Sune Darkner, Masrour Makaremi, François Chan, Peter Lampel Søndergaard, and Kenny Erleben. 2022. Open-FullJaw: An open-access dataset and pipeline for finite element models of human jaw. *Computer Methods and Programs in Biomedicine* 224 (2022), 107009.
- [9] Yixin Hu, Teseo Schneider, Bolun Wang, Denis Zorin, and Danièle Panozzo. 2020. Fast tetrahedral meshing in the wild. *ACM Transactions on Graphics* 39, 4 (Aug. 2020). doi:10.1145/3386569.3392385
- [10] Zizhou Huang, Max Paik, Zachary Ferguson, Danièle Panozzo, and Denis Zorin. 2024. Orientation-aware Incremental Potential Contact. arXiv:2402.00719 [cs.GR] <https://arxiv.org/abs/2402.00719>
- [11] Shuji Isotani, Hirofumi Shimoyama, Isao Yokota, Toshiyuki China, Shin-ichi Hisasue, Hisamitsu Ide, Satoru Muto, Raizo Yamaguchi, Osamu Ukimura, and Shigeo Horie. 2015. Feasibility and accuracy of computational robot-assisted partial nephrectomy planning by virtual partial nephrectomy analysis. *International Journal of Urology* 22, 5 (2015), 439–446.
- [12] Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine. 2011. Bounded Biharmonic Weights for Real-Time Deformation. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)* 30, 4 (2011), 78:1–78:8.
- [13] Alec Jacobson, Ladislav Kavan, and Olga Sorkine-Hornung. 2013. Robust inside-outside segmentation using generalized winding numbers. *ACM Trans. Graph.* 32, 4, Article 33 (July 2013), 12 pages. doi:10.1145/2461912.2461916
- [14] Zhongping Ji, Ligang Liu, and Yigang Wang. 2010. B-Mesh: A Modeling System for Base Meshes of 3D Articulated Shapes. *Computer Graphics Forum* 29, 7 (2010), 2169–2177. doi:10.1111/j.1467-8659.2010.01805.x arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2010.01805.x>
- [15] A. C. Liu and Barry Joe. 1994. Relationship between tetrahedron shape measures. *BIT Numerical Mathematics* 34 (1994), 268–287. <https://api.semanticscholar.org/CorpusID:120457540>
- [16] William E. Lorensen and Harvey E. Cline. 1987. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*. Association for Computing Machinery, New York, NY, USA, 163–169. doi:10.1145/37401.37422
- [17] Zoë Marschner, Silvia Sellan, Hsueh-Ti Derek Liu, and Alec Jacobson. 2023. Constructive Solid Geometry on Neural Signed Distance Fields. In *SIGGRAPH Asia 2023 Conference Papers* (Sydney, NSW, Australia) (SA '23). Association for Computing Machinery, New York, NY, USA, Article 121, 12 pages. doi:10.1145/3610548.3618170
- [18] Donald J. Marsh, Dmitry D. Postnov, Douglas J. Rowland, Anthony S. Wexler, Olga V. Sosnovtseva, and Niels-Henrik Holstein-Rathlou. 2017. Architecture of the rat nephron-arterial network: analysis with micro-computed tomography. *American Journal of Physiology Renal Physiology* 313 (2017), F351–F360. doi:10.1152/ajprenal.00092.2017 arXiv:<https://doi.org/10.1152/ajprenal.00092.2017>
- [19] Marek Krzysztof Misztal and Jakob Andreas Bærentzen. 2012. Topology-adaptive interface tracking using the deformable simplicial complex. *ACM Trans. Graph.* 31, 3, Article 24 (June 2012), 12 pages. doi:10.1145/2167076.2167082
- [20] Faezeh Moshfeghifar, Torkan Gholamalizadeh, Zachary Ferguson, Teseo Schneider, Michael Bachmann Nielsen, Danièle Panozzo, Sune Darkner, and Kenny Erleben. 2022. LibHip: An open-access hip joint model repository suitable for finite element method simulation. *Computer Methods and Programs in Biomedicine* 226 (2022), 107140.
- [21] Steffen Oeltze and Bernhard Preim. 2005. Visualization of vasculature with convolution surfaces: method, validation and evaluation. *IEEE Transactions on Medical Imaging* 24, 4 (2005), 540–548. doi:10.1109/TMI.2004.843196
- [22] Karran Pandey, Jakob Andreas Bærentzen, and Karan Singh. 2022. Face Extrusion Quad Meshes. In *ACM SIGGRAPH 2022 Conference Proceedings* (Vancouver, BC, Canada) (*SIGGRAPH '22*). Association for Computing Machinery, New York, NY, USA, Article 10, 9 pages. doi:10.1145/3528233.3530754
- [23] Athina Panotopoulou, Elissa Ross, Kathrin Welker, Evelyne Hubert, and Géraldine Morin. 2018. *Scaffolding a Skeleton*. Springer International Publishing, Cham, 17–35. doi:10.1007/978-3-319-77066-2
- [24] Dmitry D Postnov, Donald J Marsh, Dmitry E Postnov, Thomas H Braunstein, Niels-Henrik Holstein-Rathlou, Erik A Martens, and Olga Sosnovtseva. 2016. Modeling of kidney hemodynamics: probability-based topology of an arterial network. *PLOS computational biology* 12, 7 (2016), e1004922.
- [25] Johannes Reichold, Marco Stampanoni, Anna Lena Keller, Alfred Buck, Patrick Jenny, and Bruno Weber. 2009. Vascular Graph Model to Simulate the Cerebral Blood Flow in Realistic Vascular Networks. *Journal of Cerebral Blood Flow & Metabolism* 29, 8 (2009), 1429–1443. doi:10.1038/jcbfm.2009.58 arXiv:<https://doi.org/10.1038/jcbfm.2009.58> PMID: 19436317
- [26] Scott Schaefer, Tao Ju, and Joe Warren. 2007. Manifold Dual Contouring. *IEEE Transactions on Visualization and Computer Graphics* 13, 3 (2007), 610–619. doi:10.1109/TVCG.2007.1012
- [27] Jonathan Richard Shewchuk. 1998. Tetrahedral mesh generation by Delaunay refinement. In *Proceedings of the Fourteenth Annual Symposium on Computational Geometry* (Minneapolis, Minnesota, USA) (SCG '98). Association for Computing Machinery, New York, NY, USA, 86–95. doi:10.1145/276884.276894
- [28] Hang Si. 2015. TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator. *ACM Trans. Math. Softw.* 41, 2, Article 11 (feb 2015), 36 pages. doi:10.1145/2629697
- [29] Olga Sorkine and Marc Alexa. 2007. As-Rigid-As-Possible Surface Modeling. In *Proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing*, 109–116.
- [30] Gabriel Taubin. 1995. Curve and surface smoothing without shrinkage. *Proceedings of IEEE International Conference on Computer Vision* (1995), 852–857. <https://api.semanticscholar.org/CorpusID:10078417>
- [31] Chenglong Wang, Holger R Roth, Takayuki Kitasaka, Masahiro Oda, Yuichiro Hayashi, Yasushi Yoshino, Tokunori Yamamoto, Naoto Sassa, Momokazu Goto, and Kensaku Mori. 2019. Precise estimation of renal vascular dominant regions using spatially aware fully convolutional networks, tensor-cut and Voronoi diagrams. *Computerized Medical Imaging and Graphics* 77 (2019), 101642.
- [32] Brian Wyvill, Andrew Guy, and Eric Galin. 1999. Extending the CSG Tree. Warp-Blending and Boolean Operations in an Implicit Surface Modeling System. *Computer Graphics Forum* 18, 2 (1999), 149–158. doi:10.1111/1467-8659.00365 arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-8659.00365>
- [33] Peidi Xu, Niels-Henrik Holstein-Rathlou, Stinne Byrholdt Søgaard, Carsten Gundlach, Charlotte Mehlin Sørensen, Kenny Erleben, Olga Sosnovtseva, and Sune Darkner. 2023. A hybrid approach to full-scale reconstruction of renal arterial network. *Scientific Reports* 13, 1 (2023), 7569.
- [34] Peidi Xu, Faezeh Moshfeghifar, Torkan Gholamalizadeh, Michael Bachmann Nielsen, Kenny Erleben, and Sune Darkner. 2022. Auto-segmentation of Hip Joints Using MultiPlanar UNet with Transfer Learning. In *Medical Image Learning with Limited and Noisy Data*, Ghada Zamzmi, Sameer Antani, Ulas Bagci, Marius George Linguraru, Sivararam Krishnan Rajaraman, and Zhiyun Xue (Eds.). Springer Nature Switzerland, Cham, 153–162.
- [35] Peidi Xu, Olga Sosnovtseva, Charlotte Mehlin Sørensen, Kenny Erleben, and Sune Darkner. 2023. Extremely weakly-supervised blood vessel segmentation with physiologically based synthesis and domain adaptation. arXiv:2305.17054 [eess.IV]

Received 23 January 2025; revised 12 March 2025; accepted 5 June 2025

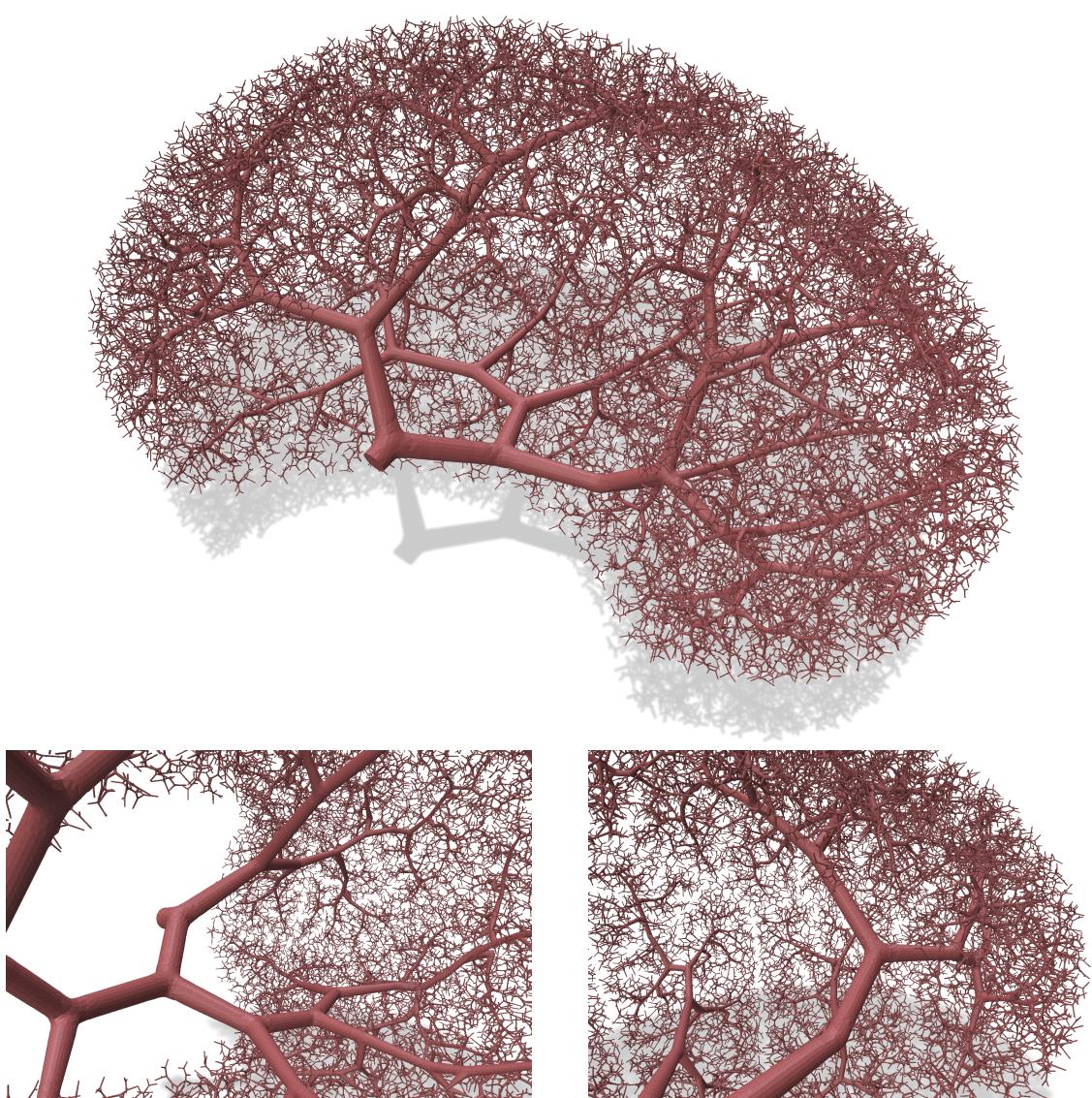


Fig. 9. Full scale volume mesh of rat renal vasculature, with 3 angles shown. Generated using resolution parameter $\rho = 4$.

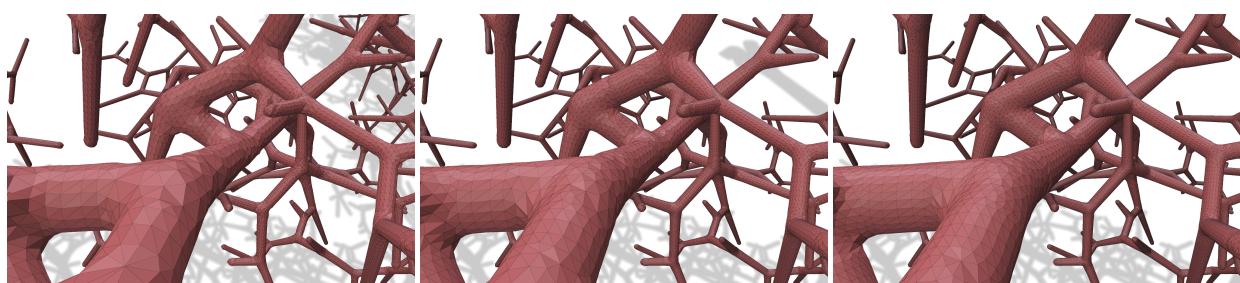


Fig. 10. Resolution parameter comparison. From left to right; $\rho = 3$, $\rho = 4$ and $\rho = 5$.

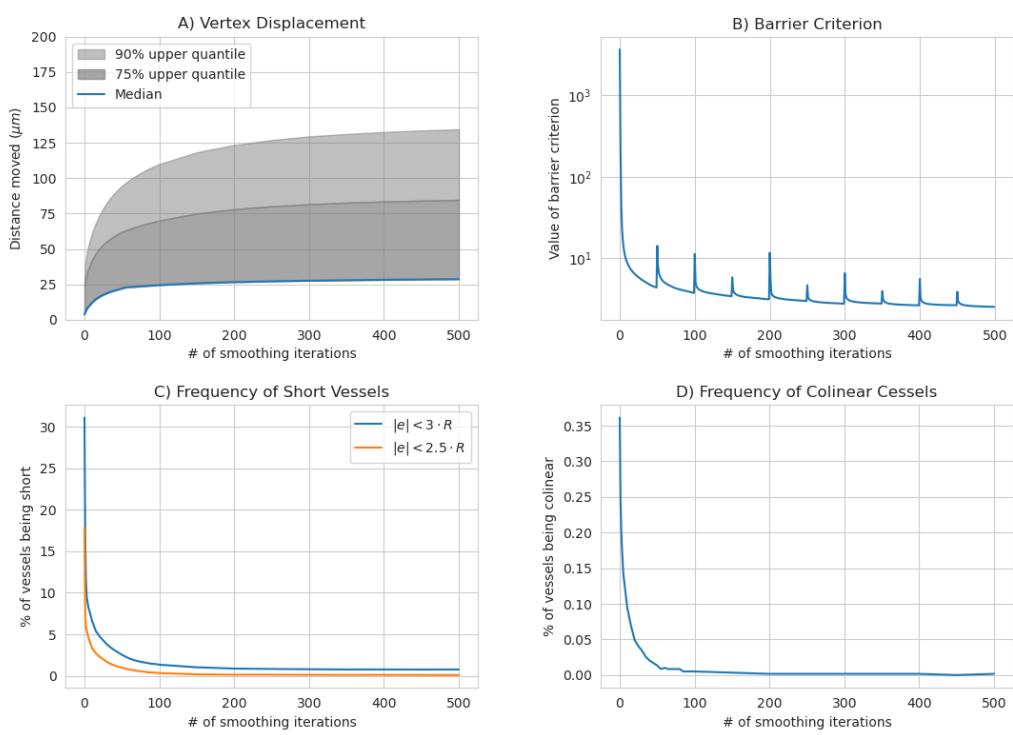


Fig. 11. Progression of skeleton metrics and node displacements as functions of the number of smoothing iterations. Only slight movement ($\sim 25\mu\text{m}$) in most nodes result in drastic improvements in the metrics. The spikes in B) is a result of the delayed recalculation of AABB overlaps. Figures C) and D) show that most but not all vessels are made *cuttable*.

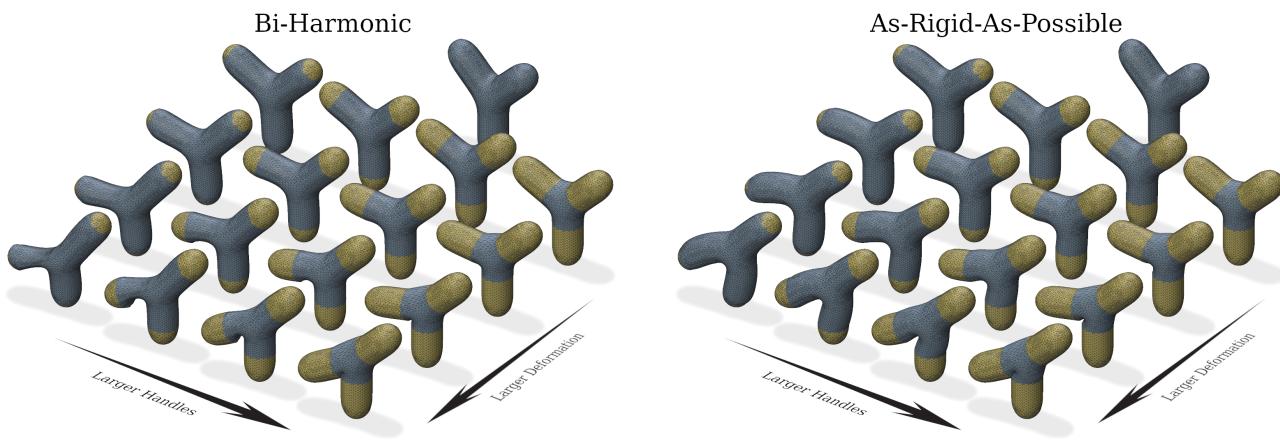


Fig. 12. Parameter studies of bi-harmonic and ARAP deformations for vessel meshes. The size of the deformation handles have significant impact on the types of geometries produced, with small handles resulting in more bent geometries, and large handles increasing the prevalence of creases. Both deformation techniques produce smooth results at small deformations but become more warped as the bend is increased.