

Image-To-Mesh Conversion for Biomedical Simulations

Fotis Drakopoulos*, Kevin Garner, Christopher Rector,
Nikos Chrisochoides*

Center for Real-Time Computing, Department of Computer Science,
Old Dominion University, Norfolk, VA, United States of America.

*Corresponding author(s). E-mail(s): fdrakopo@gmail.com;
npchris@gmail.com;

Contributing authors: kgarn006@odu.edu; chrisrector14@gmail.com;

Abstract

Converting a three-dimensional medical image into a 3D mesh that satisfies both the quality and fidelity constraints of predictive simulations and image-guided surgical procedures remains a critical problem. Presented is an image-to-mesh conversion method called CBC3D. It first discretizes a segmented image by generating an adaptive Body-Centered Cubic (BCC) mesh of high-quality elements. Next, the tetrahedral mesh is converted into a mixed-element mesh of tetrahedra, pentahedra, and hexahedra to decrease element count while maintaining quality. Finally, the mesh surfaces are deformed to their corresponding physical image boundaries, improving the mesh's fidelity. The deformation scheme builds upon the ITK open-source library and is based on the concept of energy minimization, relying on a multi-material point-based registration. It uses non-connectivity patterns to implicitly control the number of extracted feature points needed for the registration and, thus, adjusts the trade-off between the achieved mesh fidelity and the deformation speed. We compare CBC3D with four widely used and state-of-the-art homegrown image-to-mesh conversion methods from industry and academia. Results indicate that the CBC3D meshes (i) achieve high fidelity, (ii) keep the element count reasonably low, and (iii) exhibit good element quality.

Keywords: Medical imaging, Image-to-mesh conversion, Segmentation, Mesh generation

1 Introduction

Image-to-mesh conversion algorithms are widely used for the quantitative analysis of patient-specific images using the Finite Element (FE) method. This has significant implications in many areas, such as image-guided therapy [1–4], interactive surgery simulation for training young clinicians [5], and endovascular flow diversion for aneurysm geometries [6–8]. The FE method is essential in modeling tissue deformation for these applications. An intrinsic difficulty of generating meshes from isosurface-based data (i.e., parametric surfaces/volumes, level-sets, segmented multi-labeled images) is the processing and recovery of the object geometry. General-purpose mesh generators (for solid and geometric modeling applications) expect that the object boundary is either parameterized (i.e., it is defined using constructive solid geometry primitives) or explicitly defined (e.g., through the boundary discretization) as a collection of patches. To convert the isosurface data into a tetrahedral mesh, one can either (i) recover the parametrized object surface and follow up with a conventional mesh generation technique, or (ii) use a mesh generation method, which operates directly on this data type.

We propose an image-to-mesh conversion method that has been tested in generating anatomically correct models of cerebral aneurysms (to be used in predicting the efficacy of endovascular flow diversion designs) and in creating Arteriovenous Malformation (AVM) models for surgical simulations. Medical surgical simulations are classified into two main categories - predictive and interactive. A predictive simulation predicts and optimizes the outcome of an intervention by using patient-specific, pre-operative image data. Predictive simulations require high geometric, topologic, and material fidelity, where mesh generation likely takes place offline for the intervention. On the other hand, interactive simulations offer training within virtual environments to allow surgical residents to achieve certain skill levels without imposing risks on patients. Interactive simulations involved with deforming solids are widely used in physically-based simulations. In particular, the design and development of surgical simulations require accurate and efficient simulation tools due to the real-time computation requirements and physics fidelity. However, because of the complexity of the problem, these simulations are computationally intensive, making interactive and real-time constraints a challenging problem. Broadly, the main components for most interactive medical simulators require knowledge in the following areas: biomechanical and anatomical modeling, collision detection, haptics, and visualization. A virtual anatomical model constitutes a precise computerized description of a human organ, commonly generated from medical images, like Magnetic Resonance Imaging (MRI) or Computed Tomography (CT). For example, an AVM is a pathology containing draining vessels, feeding arteries, the nidus of vessel bundles, and surrounding structures. The anatomical model should contain the minimal elements to describe the small vessel features. We focus on mesh generation requirements of predictive and interactive simulations - fidelity and quality. Both smoothness and the real-time requirement will be addressed elsewhere in the future. Fidelity concerns the degree to which the mesh surface aligns with an image boundary. Quality is determined by the shape and size of elements, as this also affects the accuracy of solutions for Computational Fluid Dynamics (CFD) simulations [9, 10].

Image-to-mesh (I2M) conversion is an inherently challenging problem as it must balance trade-offs between several criteria: (i) minimizing the mesh size and maximizing the element quality by utilizing proper mesh gradation and tissue-specific multi-resolution, (ii) without sacrificing geometric and topologic fidelity, (iii) while maintaining a smooth surface to reflect a certain degree of visual reality, and (iv) simultaneously allowing the upstream numerical computations for collision detection and biomechanical FE analysis to be completed within the time constraints imposed by a surgical simulation.

We present a method called CBC3D that directly converts segmented multi-labeled image data into adaptive multi-tissue isotropic tetrahedral meshes. This method builds upon previous work [5, 11, 12], maintaining the ability to generate meshes of good quality that were shown to enhance non-rigid registration solver performance and reduce error when compared to other image-to-mesh conversion methods [10]. CBC3D initially relabels image voxels to manage image noise and eliminate non-manifold voxel connectivity. It then discretizes the segmented, labeled image with a uniform BCC lattice of high-quality tetrahedra. Red-green templates subdivide the lattice near the segmented boundaries while ensuring mesh conformity (i.e., manifold connectivity between mesh regions representing different tissues). Finally, the generated surfaces are deformed to their corresponding tissue boundaries to improve fidelity while maintaining quality.

The presented procedure focuses on several of the above simulation requirements, offering the following: (i) It provides an accurate geometric and topological representation of the medical cases in our evaluation, (ii) it provides material-dependent mesh resolution to reduce element count (i.e., fidelity can be specified per material, which will determine the level of adaptivity near mesh boundaries, allowing for a localized refinement in materials of interest) (iii) it maintains good element quality during mesh deformation, (iv) it further reduces memory and CPU requirements for the solver by introducing mixed elements, and (v) it improves the overall reliability and portability of the code as it builds upon the ITK open-source, cross-platform system¹. A single-material version of CBC3D is available within the 3D Slicer² package for visualization and image analysis. A previous multi-material version of CBC3D [5] is integrated within an interactive simulator for neurosurgical procedures involving brain AVM developed in SOFA³, a framework for real-time medical simulations.

2 Related Work

Several methods use medical data as input to create either isotropic or anisotropic 2-dimensional (2D) or 3D meshes. 2D mesh generation from medical imaging data [13, 14] is outside the scope of this paper. We focus on 3D medical images and 3D meshes. For a comprehensive overview of these 3D methods, see [15]. Anisotropic mesh generation methods [16, 17] are well suited for capturing directional information and high curvature geometries (e.g., hemodynamics). In certain cases, anisotropy offers an improvement over isotropy due to its lower number of (high aspect ratio) elements that provide high accuracy for a segmented image, compared to isotropic meshes, which

¹<https://itk.org>

²<https://www.slicer.org>

³<https://www.sofa-framework.org>

may require more elements. Anisotropic meshing techniques are also suitable for flow problems like modeling cerebral aneurysms for fluid-structure interaction (FSI) simulations [18]. However, this particular method processes geometries generated from the medical images elsewhere and do not specifically process image data. Qualitative and performance evaluations of sequential methods that process surface meshes (also generated elsewhere from medical image data) as input can be found in [19], with regards to suitability for surgical simulations and non-rigid registration applications [20]. Although CBC3D generates isotropic meshes, it reduces mesh size (element count) by converting its tetrahedral meshes into mixed-element meshes (made of tetrahedra, pentahedra, and hexahedra). CBC3D is tested on both CT and MRI data, where different types of isotropic methods like Advancing Front [21] or Terminal-Edge Bisection Methods [22, 23] were considered.

Some 3D isotropic methods utilize a Delaunay refinement scheme [17, 24–32]. In our evaluation, presented in section 4, we compare two such methods to CBC3D - CGAL’s 3D Mesh Engine⁴ and Parallel Optimistic Delaunay Method [30, 33–37] (referred to throughout this paper in short as PODM). The code prototype for CGAL, including its initial design and specifications, is presented in [28]. CGAL first constructs a boundary mesh from the image and then generates a volume mesh given the boundary mesh as input. Before refinement, a mechanism of protecting balls is set up on 1-dimensional features, if any, to ensure a fair representation of those features in the mesh. This mechanism also guarantees the termination of the refinement process, independently of the input geometry. The criteria concerning the size or shape of mesh cells and surface facets drive the refinement process. It terminates when no more mesh cells or surface facets violate the criteria. A mesh optimization phase follows the refinement [29] that attempts to remove slivers.

On the other hand, PODM constructs the isosurface of the biological object with geometric and topological guarantees while simultaneously providing good-quality surface and volume elements. PODM implements a tightly coupled, shared-memory, parallel speculative execution approach using carefully designed contention managers, load balancing, synchronization, and optimization schemes. In [32], the robustness of PODM was extended to 4D segmented images, providing faithful geometric and topological approximations (when the hyper-surface is a closed smooth manifold). However, it suffered in runtime since its space-time meshing method is sequential and requires increased memory space as opposed to the 3D meshing method. A problem with Delaunay refinement is that almost flat tetrahedra, called slivers, can survive known heuristics to remove them [24, 29, 38]. Sliver removal for Delaunay-based methods is still an open problem. On the other hand, CBC3D utilizes a lattice-based approach to generate high-quality elements rather than a Delaunay scheme.

A large number of meshing methods are based on lattice space-tree (octree) decomposition [39–51], which provides adaptively-sized, high-quality elements for several medical applications. Two of these methods, CLEAVER [39] and Lattice De-refinement (LD) [44], are compared to CBC3D in our evaluation. CLEAVER first constructs a high-quality Body-Centered Cubic (BCC) lattice that covers the input image. Then, it recovers the surface by finding the points where the lattice intersects the object.

⁴<https://doc.cgal.org/4.5.2/Manual>

Subsequently, it warps the mesh vertices to the model surface or inserts new vertices on the surface. The resultant mesh quality depends on the thresholds used to determine whether to warp or insert a vertex. Lattice De-refinement (LD) applies a mesh decimation step to an octree-based mesh to improve the element shape and/or modify the mesh size. This method allows for guaranteed bounds on the smallest dihedral angle and the distance between the segmented surface and the internal tissues' (materials') boundaries. The number of tetrahedra in the mesh is as few as possible, provided the quality and distance requirements are satisfied. Instead of performing a decimation step, other approaches improve element quality by optimizing an objective function based on tetrahedral shape measures, such as the Dual Contouring technique [46–49, 52]. Additionally, some methods warp the surfaces of the lattice to the boundaries of the object using either a mass-spring system, an FE constitutive model, or an optimization scheme [12, 26, 51].

CBC3D builds upon previous work [11, 12, 51]. The presented method extends earlier work that was used and tested only on fairly regular multi-material geometries. Both the current extension and earlier work employ a physics-based mesh deformation scheme to warp the surfaces of a Body-Centered Cubic (BCC) mesh to the physical image boundaries. The deformation scheme optimizes for fidelity and quality. Additional improvements for the reliable and accurate representation of multiple materials and their corresponding resolutions were presented in [5], an earlier version of CBC3D. The current version of CBC3D improves the geometric and topologic accuracy of the methods presented in [5, 11, 12]. Lattice subdivision uses a Euclidian Distance Transform (EDT) [53] instead of a segmented image, allowing smoother refinement within a threshold from the segmented image boundaries. The method eliminates non-manifold voxel connectivities in the segmented image to provide a more accurate construction of the object. When the lattice subdivision is complete, the disconnected mesh regions or non-manifold mesh connectivities are detected, and the method subdivides the lattice further (locally or globally) to resolve the image features.

The presented method also provides a material-dependent lattice subdivision to focus only on materials of interest, thus reducing the number of generated elements. Alternatively, global subdivision criteria can be specified. A physics-based deformation scheme eliminates poorly shaped elements with heuristics and quality criteria, such as a minimum dihedral angle or a scaled Jacobian metric. The trade-off between mesh fidelity and deformation time is balanced with the extraction of different amounts of image features needed for deformation. Mixed elements, including tetrahedra, pentahedra, and hexahedra, are introduced to reduce the number of vertices in the adaptive isotropic tetrahedral mesh. The evaluation shows that a mixed mesh can reduce the number of vertices by up to 30% without compromising the tetrahedral mesh's fidelity. Finally, this work utilizes parallel computing (for some compute-intensive kernels) to improve overall performance and allow for faster processing of large data (e.g., micro-CT imaging of stents). Although multi-threading is used to improve the performance of the method, CBC3D does not qualify (yet) as a parallel mesh generation method.

Section 3 describes the CBC3D method in detail. We evaluate CBC3D in section 4 and compare it to four common image-to-mesh conversion methods found in industry

and academia: CGAL’s 3D Mesh Engine⁵ (v4.5.2), CLEAVER [39] (v1.5.4), Lattice De-refinement (LD) [44], and PODM [30, 31]. We provide a discussion and address future work to further improve CBC3D in section 5. We finally conclude in section 6.

3 Method

CBC3D uses a segmented, multi-labeled image as input and creates an adaptive tetrahedral or mixed element mesh as an output. Subsection 3.1 describes the segmentation algorithm used to obtain the labeled image. Subsection 3.2 describes pre-processing algorithms used to improve the quality of the input image. Subsection 3.3 describes the generation of the adaptive BCC lattice. Subsection 3.4 describes the conversion of the adaptive tetrahedral mesh into a mixed element mesh. Subsection 3.5 describes mesh deformation.

3.1 Segmentation

The level set segmentation algorithm for the vessel structures uses the Vascular Modeling Toolkit (VMTK) [54]. Level sets are a kind of deformable model in which the deformable surface is represented by a 3D function whose contour at level zero is the surface in question. To extract the surface from the image, the output image is run through Marching Cubes [55] with level zero. A fast marching initialization is used, consisting of placing a set of seeds and a set of targets in the image. A front is then propagated from the seeds until the first target is met, at which point the region covered by the front is the initial deformable model. This type of initialization is effective when it is necessary to segment round objects such as aneurysms (e.g. by simply placing one seed at the center and one target on the wall, the volume will be initialized). The output segmented image contains labels, where each one represents a single tissue (Figure 2).

The Materialise Mimics⁶ software is used to perform image segmentation on the micro-CT data of commercially available neurovascular stents [6]. Two different labels are tagged onto the lumen of the sidewall aneurysm model and the flow diverter. During segmentation, the device is cropped to only retain the portion providing neck coverage of the aneurysm (Figure 2).

3.2 Image pre-processing

The previous work [5, 11, 12] re-samples the segmented image to an isotropic unit-spaced image to avoid the transformation of the index to physical coordinates and vice versa. This approach is sufficient for fairly regular geometries and improves the speed of the mesh generation. However, in the case of complex data (i.e., AVM), image down-sampling can deteriorate the quality of the segmentation and may result in disconnected image regions or non-manifold connectivity (voxels that are connected to each other via an edge or a vertex). Figure 3 illustrates such an example. The presented method does not perform image down-sampling.

⁵<https://doc.cgal.org/4.5.2/Manual>

⁶<http://www.materialise.com/en/medical/software/mimics>

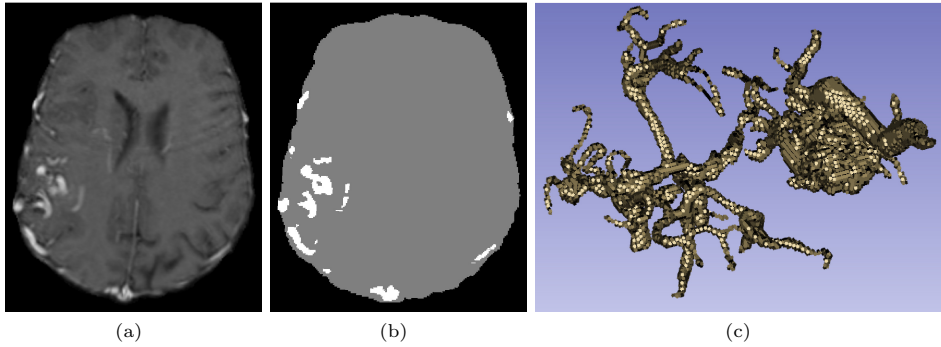


Fig. 1 Brain MRI slice with AVM after skull stripping [5] (a) and after segmentation (b). (c) depicts a volume rendering of the segmented AVM. The segmented image has a spacing of $0.7 \times 0.7 \times 1.6 \text{ mm}^3$ and a size of $320 \times 320 \times 100 \text{ voxels}^3$.

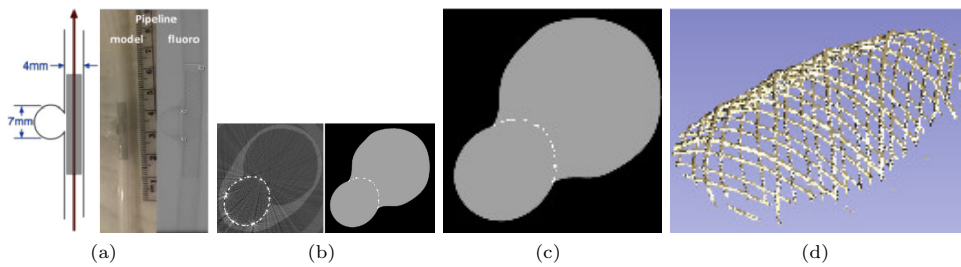


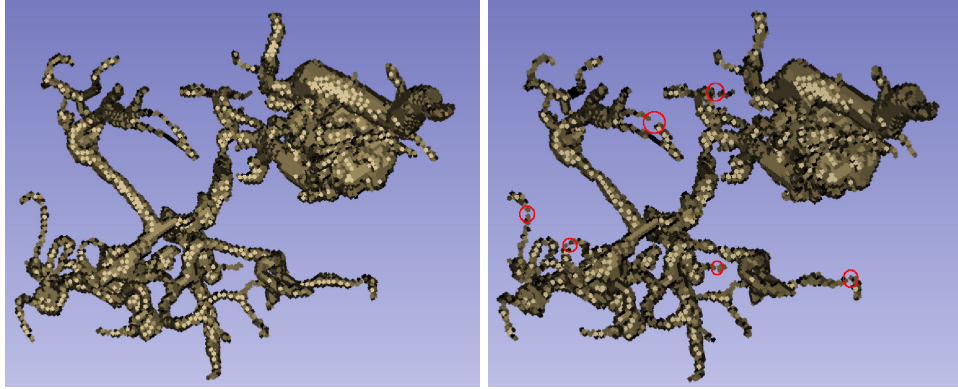
Fig. 2 Raw micro-CT slice for pipeline stent model [7] (a) before (b) and after segmentation with the device wires cropped (c). (d) depicts a volume rendering of the segmented stent. The segmented image has a spacing of $0.012 \times 0.012 \times 0.024 \text{ mm}^3$ and a size of $1001 \times 1001 \times 4421 \text{ voxels}^3$.

3.2.1 Relabeling noisy voxels

A semi-automatic or manual segmentation of poor quality may contain isolated voxels that do not correspond to an anatomical image feature. CBC3D identifies two types of such voxels and relabels them based on their neighbors. The first type has a zero label (background) and all its neighbors have a non-zero label (Figure 4a). The second type has a non-zero label and all its neighbors have a different label (Figure 4b). Not all neighbors necessarily have the same label. The neighbors are checked using a 26-neighborhood region (Figure 4c). After the noisy voxels are identified, they are relabeled to one of its neighbors' labels. Either the first, second, or both types are relabeled.

3.2.2 Relabeling disconnected regions

CBC3D controls the level of refinement in each material by using a global or a material-specific fidelity (subsection 3.3). Complex segmentations (i.e, tangled bundles of abnormal and arbitrary thin blood vessels connecting arteries and veins in the brain) may contain materials with multiple disconnected regions. CBC3D implements



(a) Before down-sampling (spacing: $0.879 \times 0.879 \times 0.879 \text{ mm}^3$) (b) After down-sampling (spacing: $1 \times 1 \times 1 \text{ mm}^3$)

Fig. 3 Brain Arteriovenous Malformation (AVM) segmentation, before and after down-sampling. The red circles indicate the problematic regions after down-sampling (i.e., disconnected voxels or non-manifold voxel connectivity).

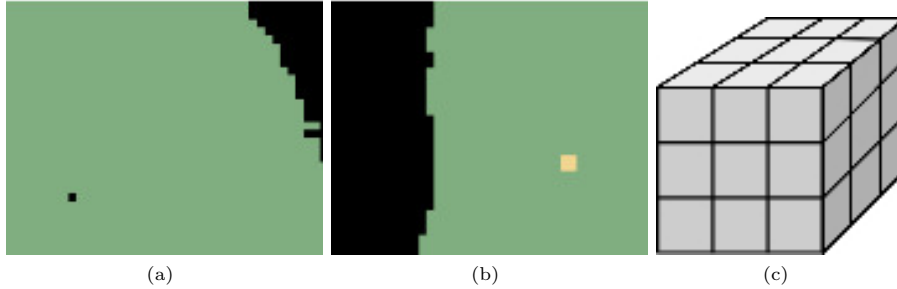


Fig. 4 Candidate isolated voxels for relabeling (a)-(b) and 26-neighborhood region (c).

a relabeling algorithm to handle each of those regions as a different material. Therefore, the present method allows for a more localized refinement which leads to a lower element count.

An ITK connected threshold image filter is employed to detect the disconnected regions in each material. A seed is first computed based on the material's label, and then small regions on the image with the same label are merged iteratively to compute a connected region. A face connectivity is used for merging. Assume that S_i is the set of voxels in material i , and the filter computes a set of voxels S_{ij} in a region j of material i . If $|S_i| = |S_{ij}|$ (where $|*|$ is the number of voxels in $*$), then material i is a single region. Otherwise, material i consists of disconnected regions. If $\frac{\|S_i\| - \|S_{ij}\|}{\|S_i\|} < s_{tol}$ then region j is too small compared to region i to be considered as tissue, hence it is relabeled to zero (background). Otherwise, region j is relabeled to a new label. A tolerance $s_{tol} = 10^{-4}$ detects the small disconnected regions. Figure 5 depicts an example before and after the relabeling process.

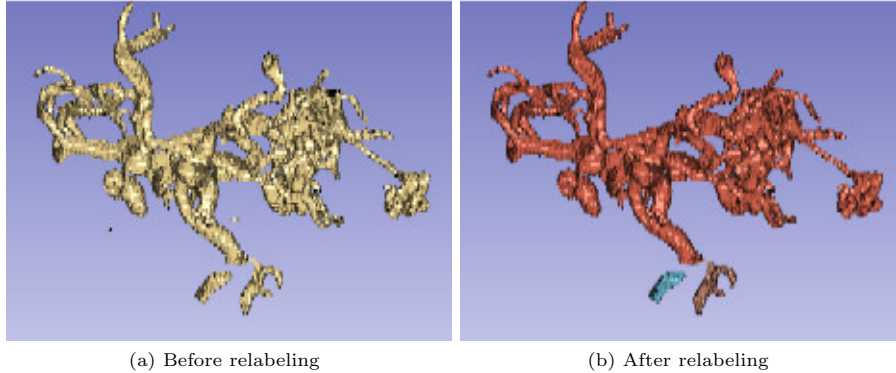


Fig. 5 Relabeling an AVM segmented image with disconnected vessels. Before processing, the image contains one material (yellow) with five disconnected regions (a). After processing, the image contains three materials (red, cyan, and brown), each of which is a single region (b). Two small disconnected regions are relabeled to a background value.

3.2.3 Eliminating non-manifold voxel connectivity

A segmentation with poor resolution may contain non-background voxels that are connected by a single edge or a vertex (Figure 7a). This type of connectivity can lead to a non-manifold mesh (i.e., tetrahedra that are connected only by a mesh edge or a mesh vertex), especially when refinement is inadequate to resolve the small image features. A non-manifold mesh deteriorates the solution accuracy in blood flow simulations within stented arterial segments or AVM surgical simulations (Figures 2, 7).

CBC3D incorporates relabeling operations with specific templates to eliminate the non-manifold voxel connectivities. First, all vertex-to-vertex connectivities are detected (Figure 6a) and eliminated by randomly selecting one of the six templates (Figures 6b-6g). All templates result in face-to-face connectivity; however, each template relabels different voxels. When all vertex-to-vertex connectivities are eliminated, the edge-to-edge connectivities are detected (Figure 6h) and eliminated by randomly selecting one of the two templates in Figures 6i-6j. Similarly, the two templates relabel different voxels. The two steps repeat until all non-manifold connectivities are eliminated. Randomness is introduced in the selection of these templates to achieve convergence in the case of relabeling neighbor clusters. Figure 7 depicts an example of an AVM segmented image before and after elimination.

3.3 Adaptive Lattice Refinement

3.3.1 BCC lattice construction

The pre-processed segmented image is discretized with a regular Body Centred Cubic (BCC) lattice. This structured lattice is generated as a result of two interlaced lattices (Figure 8). The first lattice is created by setting its vertices with an input-defined distance between them (*BCC_size* parameter in Table 3), then the edges connecting those vertices are added, consequently creating the tetrahedral elements. The elements

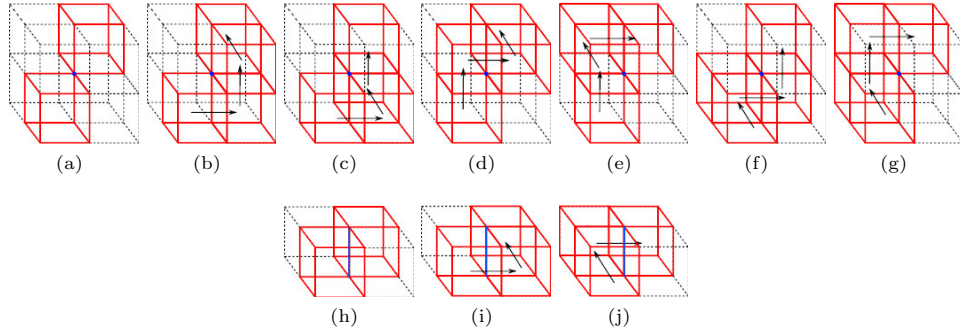


Fig. 6 Templates to eliminate a vertex-to-vertex connectivity (a) or an edge-to edge connectivity (h) in a segmented labeled image. In the case of a vertex-to-vertex connectivity, one of the six templates is randomly selected to relabel two voxels within a cluster of eight voxels (b)-(g). In the case of an edge-to-edge connectivity, one of the two templates is randomly selected to relabel a single voxel within a cluster of four voxels (i)-(j). The arrows illustrate the path of the transformation via face connected voxels after relabeling.

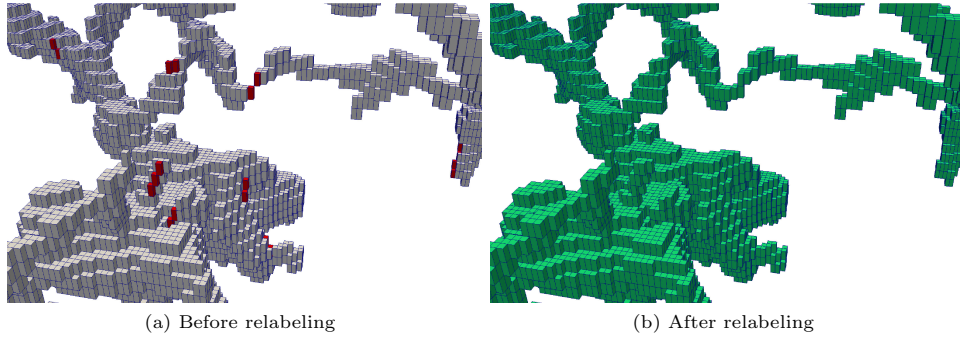


Fig. 7 AVM anisotropic segmented image ($0.7 \times 0.7 \times 1.6 \text{ mm}^3$) before (a) and after (b) relabeling to eliminate the non-manifold voxel connectivity. The voxels which are connected via an edge or a vertex are depicted with red color.

of the second lattice are created with consideration to the first lattice so that their vertices correspond to the centroids of the initial lattice cells. As a final step, the tetrahedra located completely outside of the object are discarded. The resulting elements are of the best quality possible (the minimum dihedral angle is 60°) with the regular space tiling [26].

3.3.2 Adaptive refinement

To subdivide the elements, some criterion must be considered. The current implementation uses a Euclidean Distance Transform (EDT) to decide whether an element needs to be subdivided (Figure 9). An EDT provides more information compared to a segmented image and it thus conducts a smoother refinement near the material boundaries. In the presence of n materials, n EDTs will be calculated in order to

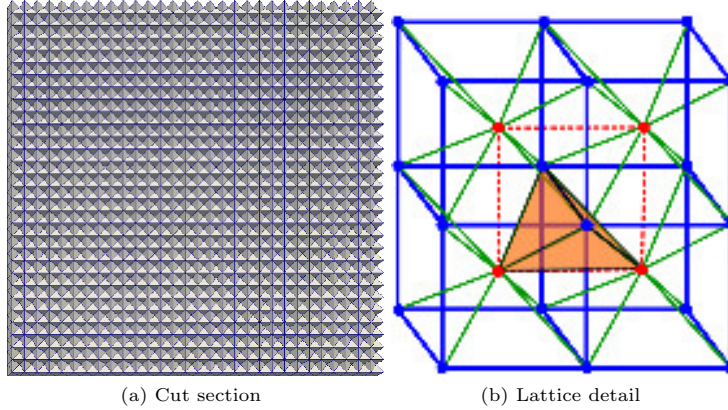


Fig. 8 Uniform Body-Centered Cubic (BCC) lattice. The green edges lace the two lattices together. Each vertex is surrounded by 14 edges and 24 tetrahedra.

exclusively subdivide the elements within each material. ITK’s Signed Maurer Distance Map Image Filter [53] is employed to compute the EDT in parallel and reduce the running time.

The lattice is refined using red (regular) - green (irregular) templates (Figure 10). An element is subdivided if the EDT to which the centroid of the element belongs changes sign in at least one of the four element vertices. Initially, all tetrahedra are marked as red. A red tetrahedron is always subdivided into eight children (1:8 regular refinement), and each child is marked red, as shown in Figure 10a. There are three choices for the internal edge of the red tetrahedron. If the shortest one is selected, the resulting eight-child tetrahedra are exactly the same as the parent except the size is one-half of the parent’s size. The red subdivision will lead to T-junctions at the newly created edge midpoints where neighboring tetrahedra are not refined to the same level. To remove the T-junctions, a green (irregular) subdivision is performed, shown in the three cases depicted in Figures 10b-10d.

To quantitatively evaluate the similarity between a mesh that corresponds to a single material (sub-mesh) and the image region of this material (sub-region), S_1 is defined as the set of all voxels in the sub-mesh, S_2 is the set of all voxels in the sub-region, and $S_1 \cap S_2$ is the point-set shared by the sub-mesh and the sub-region (common region). Ratio $F_1 = \frac{|S_1 \cap S_2|}{|S_1|}$ corresponds to the similarity between the common region and the sub-mesh, and ratio $F_2 = \frac{|S_1 \cap S_2|}{|S_2|}$ corresponds to the similarity between the common region and the sub-region. The refinement criterion is defined as: Refine the sub-mesh if $F > F_1$ or $F > F_2$, where $F \in (0, 1]$ is a global input fidelity or a material-specific fidelity (Table 3). The higher the input fidelity, the finer the mesh will be near the boundaries. The advantage of a material-specific fidelity compared to a global fidelity is that the former allows for a more localized refinement in the materials of interest, thereby reducing total element count.

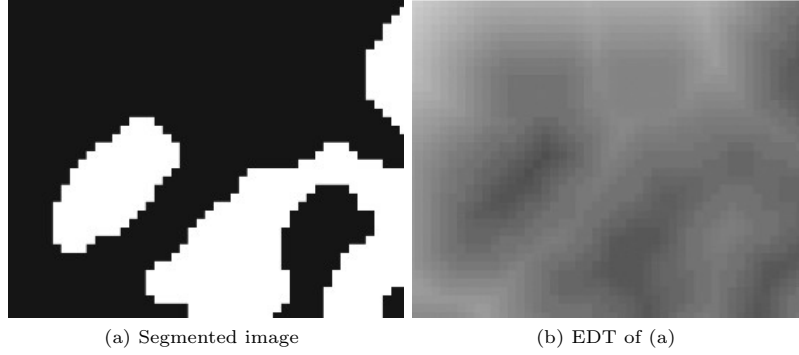


Fig. 9 Euclidean Distance Transform (EDT) computed from a labeled image. The EDT calculates the minimum distance of a voxel in the image from its closest material boundary. Voxels inside a material have a positive distance value, voxels outside a material have a negative distance value and voxels on the boundary have a zero distance value.

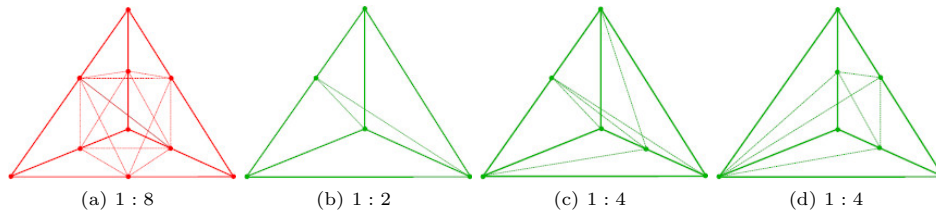


Fig. 10 Red-Green templates for lattice subdivision. 8, 2, and 4 is the number of tetrahedra after subdivision.

3.3.3 Candidate mesh selection

For a surface to be suitable for deformation, each non-background element can only be connected via at most one face with a background element. Otherwise, the elements on the surface can easily become distorted or inverted during deformation. To avoid this type of connectivity, the labels of those problematic surface elements and those of the surrounding elements are redistributed. Every tetrahedron is examined by observing the labels of its four adjacent tetrahedra. If at least three of the adjacent tetrahedra have the same label as the examined tetrahedron, then there is no need for relabeling. However, if there is more than one adjacent tetrahedron with a different label, then the examined tetrahedron needs to be relabeled to one of its neighboring labels. Between these labels, the one chosen for relabeling the tetrahedron depends on whether or not a considered label has already been examined and finalized in the previous examinations. If a neighboring label has not been fully inspected yet, the tetrahedron is relabeled to this one. At the end of the relabeling process, the background elements are discarded and the final mesh is obtained. Figure 11 depicts the basic steps of the adaptive mesh generation.

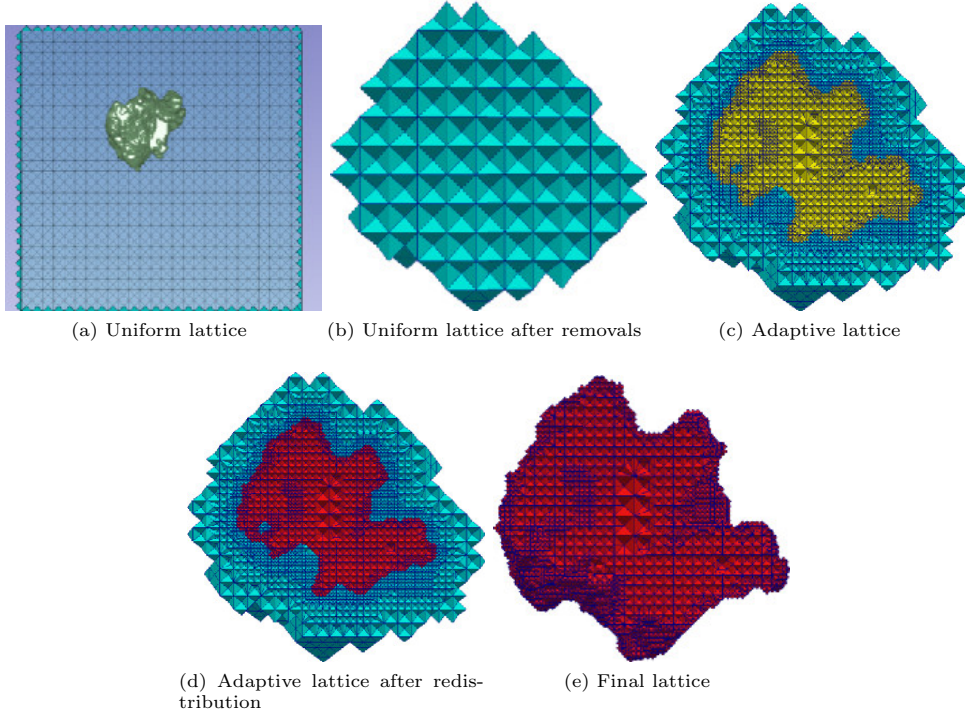


Fig. 11 Pipeline of the BCC lattice construction and adaptive refinement.

3.3.4 Mesh topological checks

The goal is to obtain a manifold topology, meaning that every tetrahedron must be connected to one another through a face. After the lattice subdivision is completed (i.e., all input fidelities are satisfied) and the candidate mesh is selected, the mesh topology is examined to ensure the absence of non-manifold (i.e. connected via an edge or a vertex) tetrahedral connections or disconnected tetrahedra in each sub-mesh.

A "mesh connected"-threshold filter was developed to detect the non-manifold topology or the disconnected tetrahedral regions in each sub-mesh. An arbitrary seed tetrahedron is first computed based on the label of each sub-mesh, and then neighbor tetrahedra are merged iteratively to compute a connected mesh region. A vertex connectivity is used for merging. The non-manifold edges and the non-manifold vertices are identified by checking the labels of the neighbor elements before merging.

When the connected threshold filter is completed, the number of tetrahedra in the connected mesh region is compared with the number of tetrahedra in the sub-mesh. If they are equal, the sub-mesh is a single connected region. However, if non-manifold topologies exist, then the tetrahedra surrounding all non-manifold vertices and all non-manifold edges are marked and they are locally subdivided in the next refinement iteration. Otherwise, the refinement of the sub-mesh is completed.

If the number of tetrahedra in the connected mesh region is not equal to the number of tetrahedra in the sub-mesh, then the sub-mesh contains disconnected regions.

In this case, all disconnected regions are first checked for relabeling. A disconnected region is relabeled if its volume is too small ($< 1\%$) compared to the volume of the sub-mesh. Relabeling is performed using a label of a neighbor region (background or not). Relabeling is canceled if it produces a non-valid manifold connectivity. If all disconnected regions are eliminated and all connectivities are manifold, then the refinement of this sub-mesh is completed. If disconnected regions exist, then the whole sub-mesh is globally refined in the next iteration. Finally, if non-manifold topologies exist, then the tetrahedra surrounding the non-manifold vertices and the non-manifold edges are marked and locally subdivided in the next refinement iteration. Although sufficient for the cases presented in our evaluation, it should be noted that this subsequent subdivision of tetrahedra surrounding non-manifold vertices and edges does not always guarantee that the sub-mesh will eventually become manifold.

The mesh topological checks should be performed only if the non-manifold voxel connectivities have been previously eliminated (subsubsection 3.2.3). Mesh topological checks can be time-consuming; therefore, they should only be employed when the application requires a manifold mesh or when the image contains small features which are harder to resolve (e.g., vessels or stents).

3.4 Mixed Element Mesh

Experimental evaluation in this study showed that a mixed mesh (with tetrahedra, pentahedra, and hexahedra) may contain up to 30% fewer vertices compared to a tetrahedral mesh of the same input, without compromising the fidelity. Therefore, it can reduce the subsequent memory and CPU requirements for the solver without any loss of accuracy.

CBC3D generates a mixed mesh from an adaptive tetrahedral lattice by merging clusters of tetrahedra into hexahedra. A valid transition between the tetrahedra and the hexahedra is guaranteed with the use of prismatic elements (i.e., pyramids). Merging is performed only within the homogeneous regions where the lattice is uniform. Therefore, the topology of the generated surfaces or interfaces between materials is preserved.

In the BCC lattice, the cardinality of a vertex in a uniform region is always 14, meaning that each vertex has 14 edges to check in the process of merging (Figure 8b). Beginning from the generated adaptive BCC mesh, the inner tetrahedral vertices are located to start the transformations. For every vertex, its adjacent tetrahedra constitute a polyhedron. To transform each one of these polyhedra to hexahedra, their inner tetrahedra must be red (according to the red-green refinement) and their labels need to be the same. If indeed all the tetrahedra of the polyhedron are of the same label and none of them are green, then for each adjacent orthogonal edge of the vertex (Figure 8b), its four attached tetrahedra are computed. Their four orthogonal edges form the quadrilateral face of the hexahedron. However, if an attached tetrahedron between these four is green or has a different label than the others, then a transition pyramid is created. This element has its base perpendicular to the orthogonal edge and by being adjacent to the hexahedral face, its vertices and edges are calculated. After creating each hexahedron, the tetrahedra inside the hexahedron are removed and the

mesh topology is updated. Figure 12 compares a tetrahedral mesh and a mixed mesh of the same input.

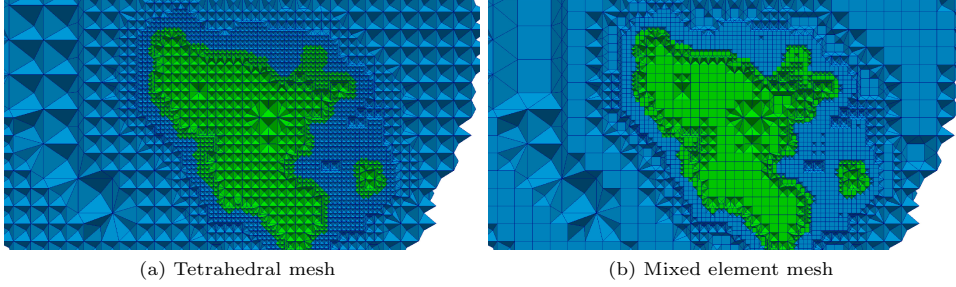


Fig. 12 Adaptive BCC lattice before (a) and after (b) it is converted into a mixed element mesh.

3.5 Mesh Deformation

Red-green templates guarantee the high quality of the adaptive mesh. Indeed, the worst minimum dihedral angle achieved in our evaluation is 30° and the worst maximum dihedral angle is 116.5° . However, the surface of this mesh can be relatively bumpy, so it may not be suitable for applications such as AVM surgical simulations or CFD modeling of brain aneurysms/stents for endovascular flow diversion. For example, a surgical simulation requires a smooth surface so that the mesh will reflect a certain degree of visual reality. A bumpy surface can deteriorate the accuracy of a CFD solution.

An approach that deforms the mesh surfaces to their corresponding image boundaries by minimizing an energy function was presented in [12]. The present study utilizes and improves upon this method in several ways. First, the new implementation is built upon the ITK toolkit, therefore improving the overall reliability and portability of the software. Second, it employs heuristics and quality control to eliminate highly distorted elements during deformation. Third, it allows meshes with mixed elements to undergo deformation. Fourth, it controls the trade-off between deformation time and mesh fidelity. Finally, it reduces deformation time using parallel computing.

This study formulates the deformation problem as an energy minimization problem represented by the objective function

$$W = U^T K U + (H U - D)^T (H U - D) \quad (1)$$

where U is the unknown mesh displacement vector, K is the mesh stiffness matrix, H is a linear interpolation matrix, and D is the vector of correspondences computed between two point sets: a source and a target point set. The source point set contains the coordinates of the surface/interface vertices of the mesh to be deformed. The target point set contains the coordinates of the voxels on the surface/interface of the materials in the image. The K matrix depends on the element type and the mechanical properties of the brain model. First, the stiffness matrix of each element

(e.g., tetrahedron, pyramid, hexahedron) is calculated in each integration point using Gaussian quadrature. The global matrix K is then assembled from the individual stiffness matrices [56].

To minimize (1), the gradient with respect to U must be zero, leading to a linear system of equations:

$$(K + H^T H)U = H^T D \quad (2)$$

An iterative solver is employed [57] to solve the system. A linear assumption is made for the material stiffness and the displacements of the biomechanical model. Once U has been found, the mesh is updated by adding the computed displacements to the current coordinates of the vertices, and the procedure is repeated until the iterations reach a maximum number. Figure 13 illustrates an example of the procedure.

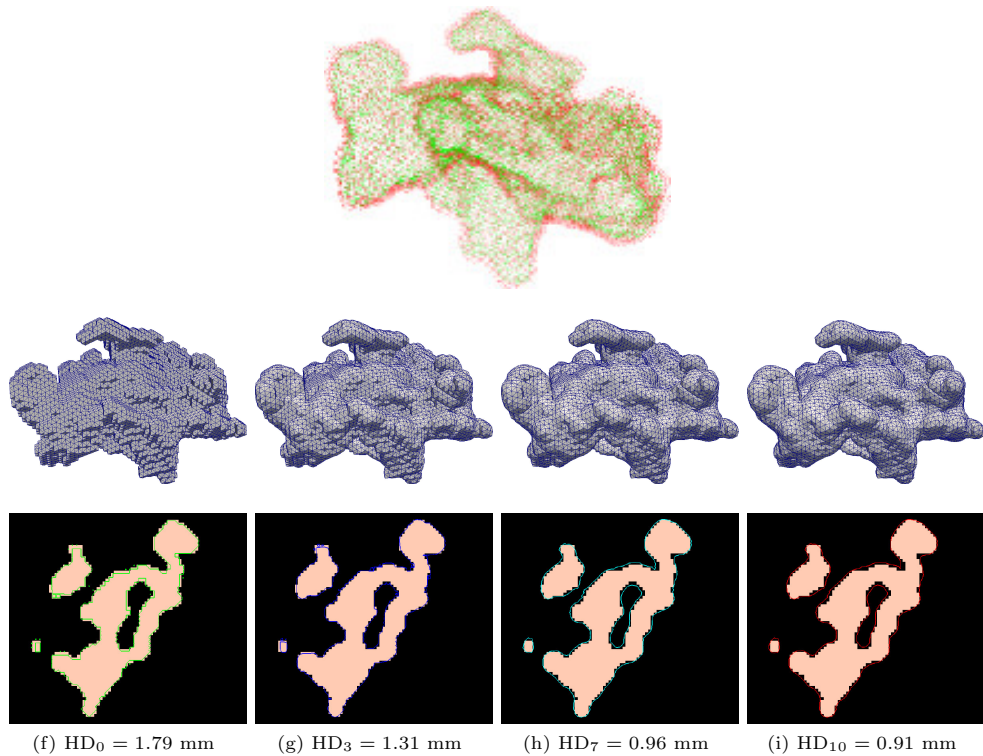


Fig. 13 Nidus mesh during deformation with 10 iterations. The figure on the top depicts the extracted source (green) and target (red) points used for deformation. Each column depicts the deformed mesh and an intersection between the mesh surface and the image plane at iterations $i = 0, 3, 7, 10$ (from left to right). HD_i denotes the mesh fidelity in terms of a Hausdorff Distance metric, at iteration i . The smaller the HD value, the higher the fidelity. As the number of iterations advances, the mesh exhibits a smoother surface.

3.5.1 Extraction of source and target points using a multicore processor

The source and the target points are extracted from the mesh to be deformed and the segmented image, respectively. The source points are the vertices on the mesh surface or on the interface between sub-meshes of different materials. The target points are the physical centers of the voxels on the segmented image boundaries.

The extraction is performed before deformation begins. For the extraction of the source points, the mesh is first decomposed into k sub-meshes so that the number of elements in each sub-mesh is approximately the same (k is the number of threads). Each thread extracts points from a single sub-mesh. The labels of the elements surrounding each source point are stored in a label set. This is because the deformation essentially registers pairs of source and target points with same label sets. For the extraction of the target points, the segmented image is first partitioned into a number of k sub-regions using an ITK threaded image region partitioner filter. Points are extracted from the segmented boundaries of each sub-region with the help of EDTs (EDTs are previously calculated at the refinement step). Additionally, the labels within a 26-neighborhood region (Figure 4c) around the target point are stored. Figure 13 depicts the extracted source and target points from a Nidus tetrahedral mesh and image, respectively.

3.5.2 Quality control

For each source point, an average displacement vector is calculated from the corresponding target points within a 3D region around the source point [12]. To improve element quality after each deformation iteration, the size of this 3D region is limited by a local element size, i.e., the average length of the element edges surrounding the source point. Then vector D is assembled from the calculated displacement vector of each source point. After each deformation iteration, a quality metric (i.e., minimum dihedral angle for tetrahedra or scaled Jacobian for hexahedra and pyramids [58]) is computed for each element. The elements that do not satisfy a minimum quality (e.g., 5° for dihedral angles or 0.2 for scaled Jacobian) are marked, and the displacement vectors of the source points surrounded by at least one marked element are scaled with a factor of 0.2. D is assembled again and the system in (2) is solved. If the quality metric is not satisfied after three consecutive attempts, the deformation stops and the procedure recovers the mesh coordinates of the previous deformation iteration. The default minimum quality parameters can be changed by the user if desired. The default values strike a balance between generating a good quality mesh and reducing mesh size (i.e., the number of elements) for the cases in our evaluation. We also determine these values based on past experience [44] and on a study involving a deep learning model that found the optimal input parameter values for an adaptive non-rigid registration application [59, 60].

3.5.3 Adjustable number of target points

The number of source points is fixed during deformation because the mesh remains topologically the same; however, the number of target points can be adjusted to

improve the accuracy of the displacement vector of a source point. Connectivity patterns are used to control the number of extracted target points. The connectivity patterns prohibit the selection of points that are too close to each other. A “vertex” (26-connectivity), an “edge” (18-connectivity), or a “face” (6-connectivity) pattern avoids the selection of neighboring voxels connected via a “vertex”, an “edge” or a “face,” respectively. The “no” option disables the connectivity patterns, hence maximizing the number of selected points. Figure 14 depicts the extracted target points using the available patterns. Table 1 presents qualitative results on mesh deformation using those patterns. The higher the number of target points, the more accurate but slower the computation. For the experiments in this study, a “face” pattern is employed to balance the trade-off between accuracy and speed.

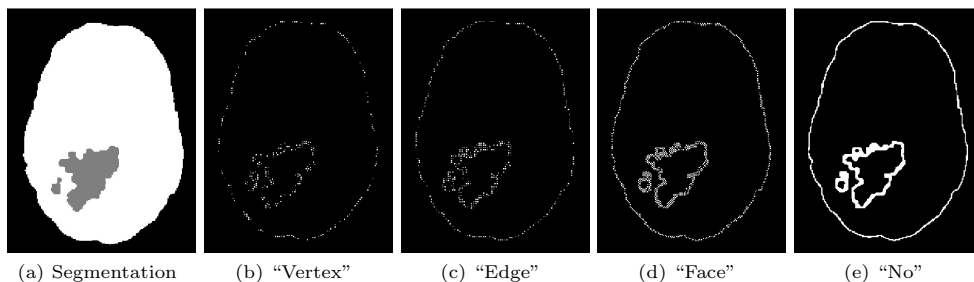


Fig. 14 Extracted target points from a brain-nidus segmented image using the available connectivity patterns. Each pattern results in a different number of points. The number of points for “Vertex,” “Edge,” “Face,” and “No” patterns is 21510, 26387, 47306, and 91906, respectively. For simplicity, the points in one volumetric slice are depicted.

Table 1 Performance of deformation for a nidus geometry (Figure 13) using the available non-connectivity patterns. HD is a Hausdorff Distance metric. HD_{10} corresponds to the mesh fidelity after a deformation step of 10 iterations. $HD_0 = 1.79$ mm is the mesh fidelity before the deformation. The smaller the HD value, the higher the fidelity. The experiment was conducted on a machine with an Intel i7-2600@3.40 GHz CPU, and 16 GB of RAM.

Pattern	#Target Points	HD_{10} (mm)	Time (sec)
“Vertex”	2770	0.97	21.01
“Edge”	3498	0.91	22.57
“Face”	6213	0.88	26.19
“No”	11755	0.84	35.27

4 Evaluation Results

The CBC3D software is evaluated on four segmented images. Table 2 lists the image data. Cases 1 and 4 are obtained from the Neurosurgery Department at Stony Brook University. Cases 2 and 3 are obtained from Kitware⁷. In case 4, the volume image

⁷<https://www.kitware.com>

is obtained after combining 4182 bitmap slices using an ITK Tile Image Filter. The experiments for cases 1-2 are performed on a DELL workstation with 8 hardware cores, Intel(R) Core(TM) i7-2600 CPU @ 3.40 GHz, and 16 GB RAM. The experiments for cases 3-4 are performed on a DELL workstation with 24 hardware cores, Intel(R) Xeon(R) CPU E5-2697v2@2.70 GHz, and 757 GB RAM.

Table 2 Segmented imaging data for experimental evaluation.

Case	Type	Materials	Image Spacing (mm ³)	Image Size (voxels ³)
1	Isotropic	Aneurysm	1.00 × 1.00 × 1.00	512 × 512 × 508
2	Anisotropic	Brain-Tumor	0.48 × 0.48 × 1.00	384 × 512 × 176
3	Anisotropic	Brain-AVM	0.70 × 0.70 × 1.60	320 × 320 × 100
4	Anisotropic	Lumen-LVIS Stent	0.012 × 0.012 × 0.024	1001 × 1001 × 4182

CBC3D is compared with four image-to-mesh conversion codes: CGAL’s 3D Mesh Engine⁸ (v4.5.2), CLEAVER [39] (v1.5.4), Lattice-Derefinement (LD) [44], and PODM [30, 31]. CGAL and CLEAVER are open-source codes. LD and PODM are codes previously developed by CRTC⁹. CBC3D, CLEAVER, and LD are lattice-based methods. CGAL and PODM are Delaunay-based methods.

Table 3 lists the parameters used for each software in our evaluation. For PODM, parameter $\delta > 0$ specifies the size of the mesh. The smaller the δ , the higher the element count. As mentioned previously, CLEAVER initially constructs a high-quality BCC lattice that covers the input image. Next, it locally warps or cuts the lattice so that it will conform to multi-material arbitrary surfaces. The violation parameters α_{short} , α_{long} decide the trade-off between snapping/warping and stencil cleaving; therefore, they implicitly control the size of the mesh. In this evaluation, the default violation parameters are used, thus a worse case minimum dihedral angle of 2.76° and worst case maximum dihedral angle of 175.42° is achieved. LD allows for guaranteed bounds on the smallest dihedral angle and on the distance between the boundaries of the mesh and the boundaries of the materials [44]. The highest possible fidelity and a minimum dihedral angle of 15° is specified for all the experiments.

Figures 15 and 16 depict cuts of the generated meshes. CLEAVER exhibits the most rapid element gradation among all the methods. CGAL, CLEAVER, and LD fail to generate a mesh for the Lumen-LVIS stent case (Table 2). CGAL’s image reader cannot read the input image. CLEAVER terminates the program with a message: "Cleaver Tet Mesher terminated with an unknown exception". LD throws an instance of `std::bad_alloc` because it exceeds the physical memory of the system (757 GB). For comparison purposes, CBC3D and PODM allocate 354 GB and 141 GB, respectively.

Tables 4 - 5 report quantitative results on element count and element quality. The lattice-based methods (CBC3D, CLEAVER, and LD) result in larger meshes compared to the Delaunay-based methods (CGAL and PODM) mainly because the templates impose stricter rules for element subdivisions. Nevertheless, the largest mesh in this study is generated by PODM (15.85 M) because a small element size ($\delta = 0.02$) is specified to resolve the LVIS stent. Figure 17 compares the meshes of the LVIS stent.

⁸<https://doc.cgal.org/4.5.2/Manual>

⁹<https://crtc.cs.odu.edu>

Table 3 Input parameters for experimental evaluation.

Method	Parameter	Value	Description
CBC3D	$lattice_{sp}$	10 mm (Cases 1-3) 0.15 mm (Case 4)	lattice spacing
	F	0.95	fidelity
	$connectivity$	“no”	pattern for target point extraction
	N_{iter}	5 (Cases 1-3) 7 (Case 4)	number of smoothing iterations
	$facet_angle$	30°	lower angle bound for surface facets
CGAL	$facet_size$	2	radii upper-bound for Delaunay balls
	$facet_distance$	2	face distance upper bound
	$cell_radius_edge_ratio$	1.5	radius-edge ratio upper bound
	$cell_size$	2.5 (Cases 1-2) 1.5 (Case 3) 0.2 (Case 4)	circumradii upper-bound
	CLEAVER	α_{short}	0.357
α_{long}		0.203	axis-aligned threshold for edge-cuts
LD	$i2m$	0	image-to-mesh distance
	$m2i$ $angle$	0 15°	mesh-to-image distance minimum dihedral angle
PODM	δ	1.2 (Cases 1-2) 0.6 (Case 3) 0.02 (Case 4)	element size

PODM does not adequately resolve the features of the stent, hence the mesh appears to be disconnected (Figure 17b).

CLEAVER and LD generate meshes with high-quality elements, although no experimental results are available for case 4 (Tables 4 - 5). LD imposes a quality bound in the generated mesh (i.e., minimum dihedral angle of 15°) by adjusting the element count appropriately. CBC3D produces meshes of well-shaped elements. Deformation does not significantly affect case 3, hence the angle extrema are similar to those in the adaptive lattice. The Delaunay-based methods generate meshes of reasonably well-shaped elements.

Table 4 Evaluation results on element count.

Case	#Tetrahedra				
	CBC3D	CGAL	CLEAVER	LD	PODM
1	272K	300K	4.33M	776K	370K
2	578K	517K	3.20M	2.54M	244K
3	5.05M	1.68M	3.59M	1.48M	1.03M
4	12.98M	-	-	-	15.85M

Figures 18-21 illustrate the element angle distribution of the meshes. The Delaunay-based methods exhibit a Gaussian distribution of dihedral angles. The lattice-based methods exhibit a different distribution due to a more structured connectivity. For example, in CLEAVER, about 41% of the dihedral angles are between $60^\circ - 65^\circ$, and about 26% are $90^\circ - 95^\circ$. In CBC3D, about 50% of the dihedral angles are between $55^\circ - 65^\circ$, and about 20% are between $85^\circ - 95^\circ$. In LD, about 15% of the angles are between $45^\circ - 50^\circ$, and about 20% are between $90^\circ - 95^\circ$.

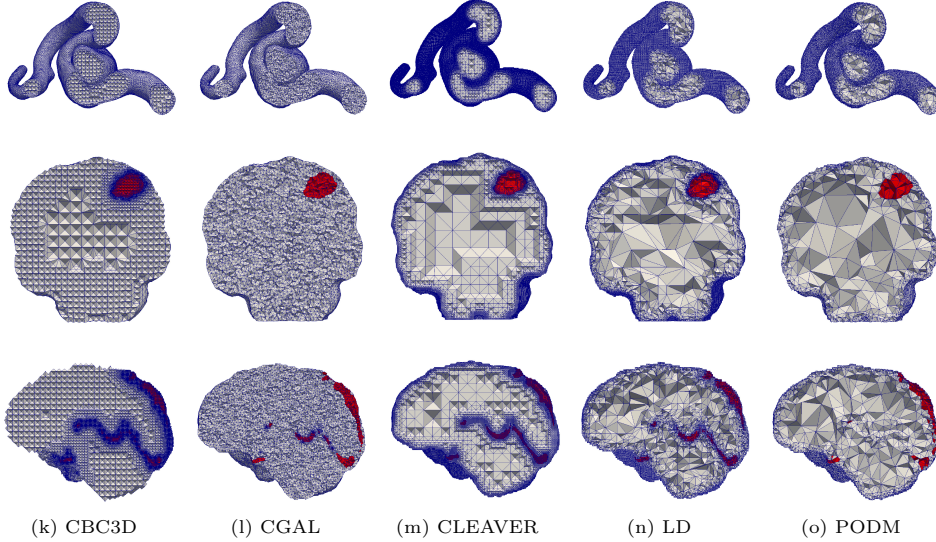


Fig. 15 Cuts of the generated tetrahedral meshes. The top, middle, and bottom row correspond to cases 1, 2, and 3, respectively. Each column depicts meshes that are generated with a single method. Identical cut section planes are used for all the meshes in a single case. The growth from small to large elements varies among the methods. The quality of these meshes is evaluated using a min/max dihedral angle metric and an element angle distribution in 5-deg increments.

Table 5 Evaluation results on element quality. The minimum and maximum dihedral angles are reported in degrees $\in (0^\circ, 180^\circ)$. The larger the minimum angle and the smaller the maximum angle, the higher the quality.

Case	Dihedral angle (min, max)				
	CBC3D	CGAL	CLEAVER	LD	PODM
1	(5.07°, 171.73°)	(12.04°, 162.23°)	(25.06°, 126.94°)	(15.00°, 171.05°)	(4.54°, 170.13°)
2	(8.89°, 166.67°)	(12.00°, 162.73°)	(11.34°, 153.15°)	(15.00°, 171.86°)	(4.90°, 170.26°)
3	(29.94°, 116.67°)	(1.41°, 176.76°)	(5.81°, 157.66°)	(15.00°, 170.55°)	(4.40°, 170.24°)
4	(4.95°, 173.10°)	-	-	-	(2.23°, 176.44°)

The mesh fidelity is qualitatively evaluated on AVM data (case 3). For this purpose, the AVM mesh is first extracted from the multi-material mesh and then superimposed on the AVM segmentation (Figure 22). The closer the mesh surface is to the boundary of the segmented AVM image, the higher the fidelity. The LD method achieves high fidelity because it completely resolves the vessels (it creates a voxelized mesh surface). Nevertheless, Figure 22e indicates a small shift in the output mesh in relation to the input image, most likely due to image resampling. CLEAVER resolves most of the vessel structures, but the generated mesh is noticeably shifted (Figure 22d). CBC3D achieves satisfactory fidelity. CBC3D's mesh topological checks are turned off to avoid further red-green subdivisions, keeping the element count low.

Quantitative evaluation on mesh fidelity is performed using a two-sided Hausdorff Distance metric: $HD = \max\{HD_{I \rightarrow M}, HD_{M \rightarrow I}\}$, where $HD_{I \rightarrow M}$ is the value of the

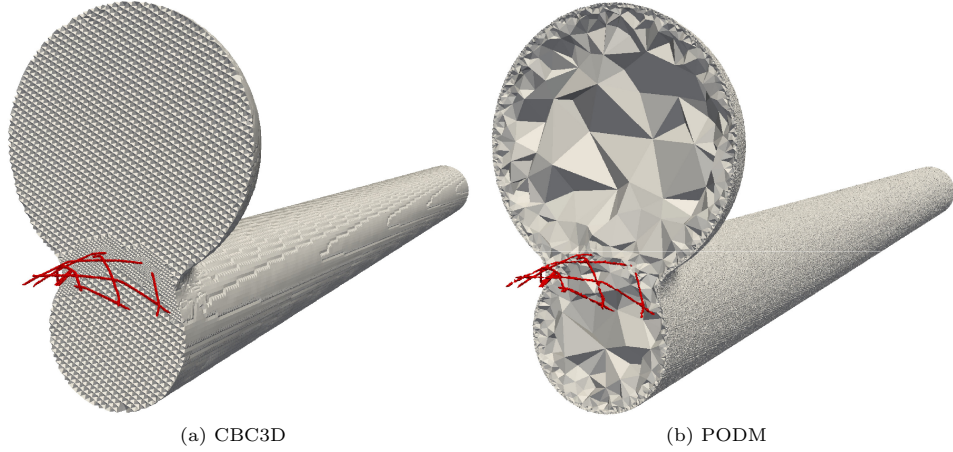


Fig. 16 Cuts of the generated tetrahedral meshes of the Lumen-LVIS Stent.



Fig. 17 Extracted mesh of the LVIS stent.

metric from the image to the mesh, and $HD_{M \rightarrow I}$ is the value of the metric from the mesh to the image. The lower the HD error, the higher the fidelity. HD is computed between two point sets. The first point set contains the coordinates of the vertices located on the mesh surface. The second point set contains the coordinates of the voxels located on the boundaries of the segmented material (i.e., where the EDT value is zero). HD is computed for each material and the maximum value is reported. An open-source implementation is employed to calculate the HD metric [61]. Tables 6 - 7 report the HD values and the end-to-end running time for mesh generation. PODM is a parallel multi-threaded code. CBC3D is sequential but computes the EDTs, source, and target points in parallel. The remaining codes are sequential. Figure 23 depicts plots of the results.

It should be noted that the error in Table 6 is relative to the image spacing. An error approximately 3-4 times the spacing may be satisfactory, depending on the application. LD achieves the highest fidelity among all the methods, but no experimental results are available for case 4. PODM and CBC3D exhibit a reasonably good fidelity.

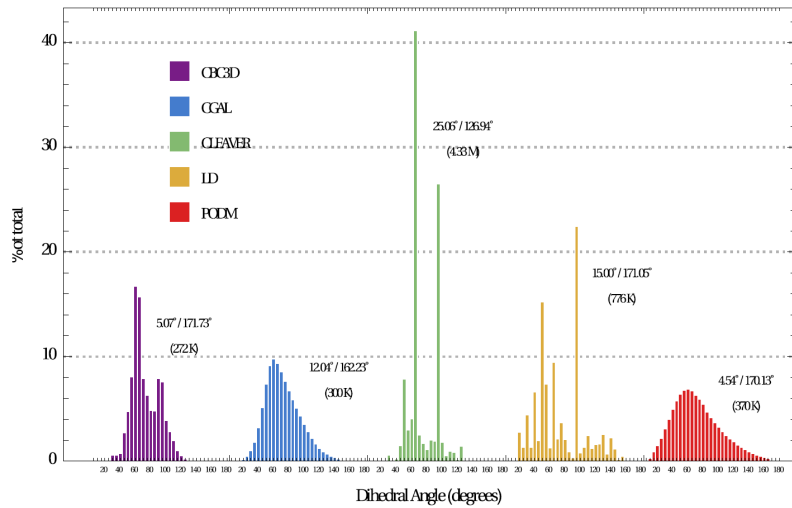


Fig. 18 Element angle distribution (in 5-deg increments) of Cavernous Aneurysm meshes. The min/max dihedral angles and the element count are reported for each method.

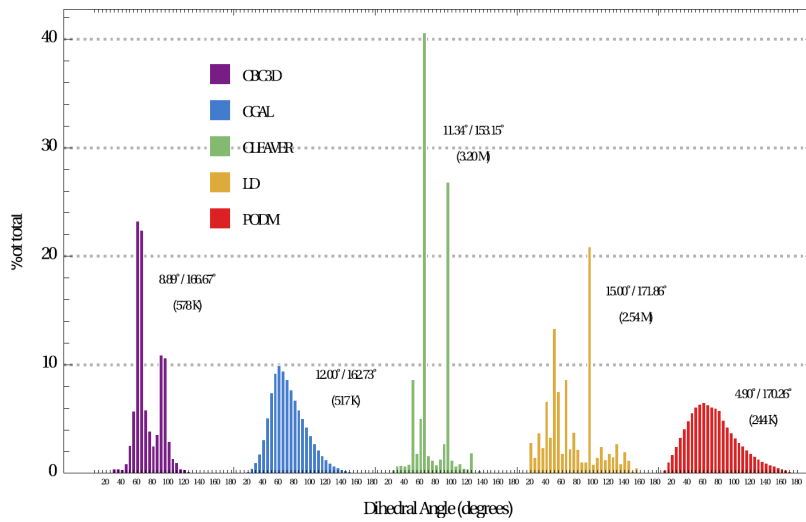


Fig. 19 Element angle distribution (in 5-deg increments) of Brain-Tumor meshes. The min/max dihedral angles and the element count are reported for each method.

A fine mesh that accurately conforms to a segmented image typically has a voxelized appearance. However, the voxelized segmentation is itself inaccurate. The true surface is smooth and a binary segmentation does not capture this fact. On the other hand, a smooth surface mesh is easier to obtain with a lower fidelity. Therefore, fidelity must be compromised in favor of a more realistic appearance. Applications such as interactive surgical simulations require volume meshes with smooth surfaces. Additionally, a bumpy surface can deteriorate the accuracy of a CFD solution.

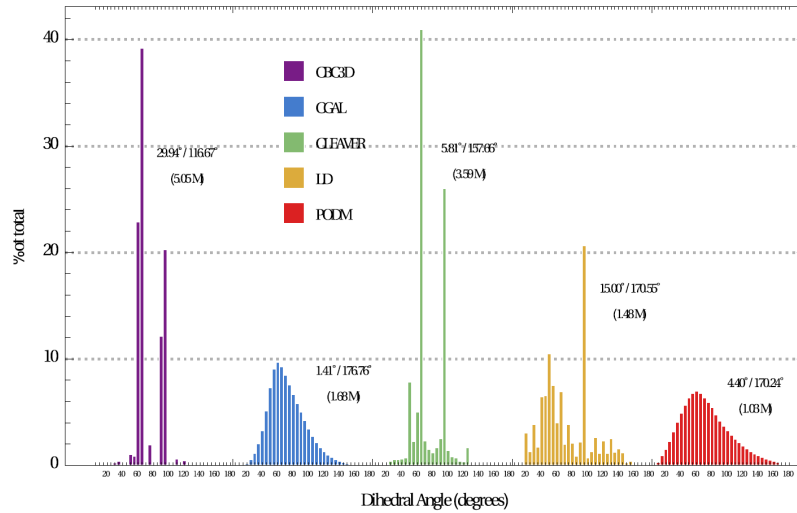


Fig. 20 Element angle distribution (in 5-deg increments) of Brain-AVM meshes. The min/max dihedral angles and the element count are reported for each method.

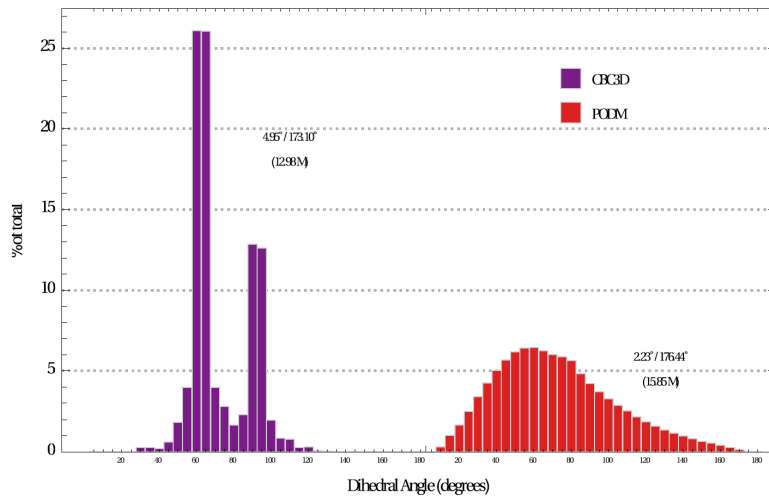


Fig. 21 Element angle distribution (in 5-deg increments) of Lumen-LVIS stent meshes. The min/max dihedral angles and the element count are reported for each method.

Figure 24 compares the surface meshes of a Cavernous Aneurysm. CBC3D exhibits the smoothest surface among all the methods. CGAL creates a relatively smooth surface as it follows a two-step approach; it first reconstructs the surface and then generates a volume mesh given the surface as an input. CLEVER and LD generate voxelized surfaces. PODM generates a bumpy surface (Figure 24).

CBC3D's tetrahedral meshes are converted to mixed meshes. Figures 25 and 26 depict the results. A mixed mesh reduces the number of mesh vertices, as well as the subsequent memory and CPU requirements for the solver without any loss of accuracy.

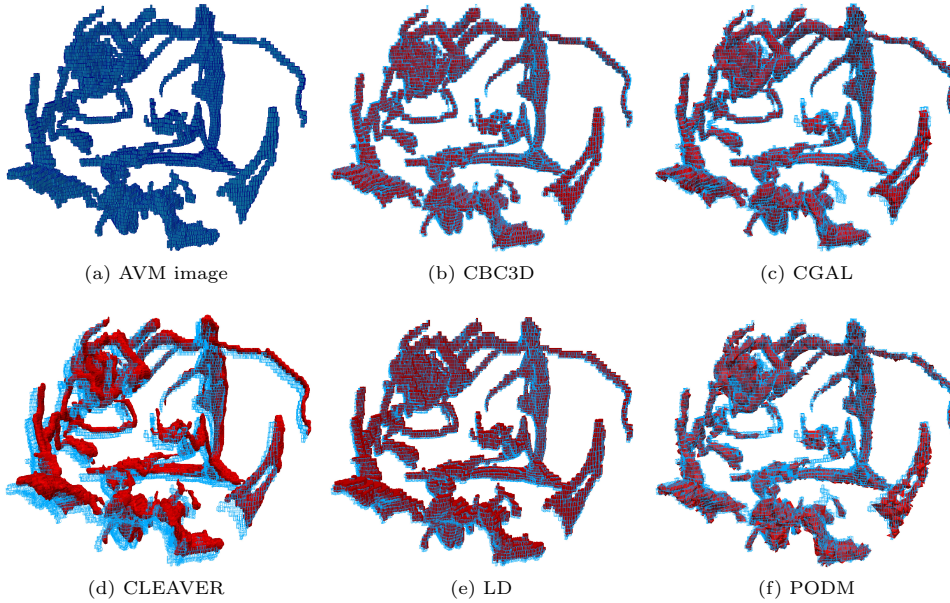


Fig. 22 Qualitative evaluation on the fidelity of AVM mesh. Figures (b)- (f) depict the AVM mesh (red) superimposed on the AVM segmentation (blue). The closer the mesh surface is to the boundary of the segmented material, the higher the fidelity.

Table 6 Results on mesh fidelity calculated using a two-sided Hausdorff Distance metric. The lower the HD value, the higher the fidelity.

Case	HD (mm)				
	CBC3D	CGAL	CLEAVER	LD	PODM
1	4.51	3.26	5.56	1.73	2.42
2	4.59	2.51	5.03	1.41	3.87
3	3.91	18.09	6.14	2.19	3.65
4	0.34	-	-	-	0.33

The number of vertices is reduced by 2.35%, 22.28%, 23.40%, and 30.22% for cases 1, 2, 3, and 4, respectively. The mixed meshes exhibit qualities and smooth surfaces similar to the adaptive tetrahedral meshes.

5 Discussion and Future Work

CBC3D noticeably does not exhibit a good element size gradation, such as that generated by Cleaver in Figure 15 or PODM in Figure 16. This is due to the specified lattice spacing parameter. We specify these values so that CBC3D can generate elements of good quality, and then maintain that quality while the mesh is deformed to improve fidelity. As mentioned previously in section 3.5.2, the 3D region in which the displacement vector from equation (1) is calculated during deformation is limited by

Table 7 Results on end-to-end running time. The time for writing the mesh is not included. T represents the number of threads utilized.

Case	Mesh Generation Time (seconds)				
	CBC3D	CGAL	CLEAVER	LD	PODM
1	68.24 (8T)	12.33	470.22	68.44	7.86 (8T)
2	101.03 (8T)	14.92	173.85	97.76	7.93 (8T)
3	384.86 (24T)	51.59	59.13	40.19	10.98 (24T)
4	3932.98 (24T)	-	-	-	750.02 (24T)

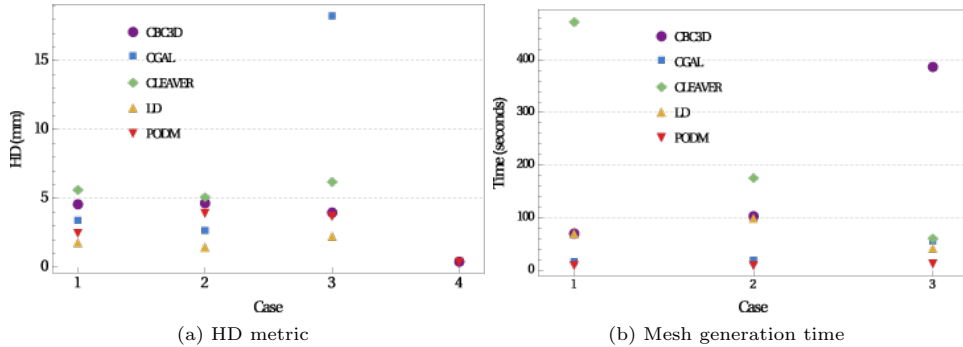


Fig. 23 Plots of the results in Tables 6 - 7. (b) does not include case 4 (3932.98, and 750.02 seconds for CBC3D and PODM, respectively).

the local element size. Additionally, we quantitatively evaluate the similarity between a mesh that corresponds to a material and the image region of this material (as specified in section 3.3.2) based on metrics also utilized in [12], which were shown to satisfy the requirements of non-rigid registration solvers for brain images [10]. We will, however, utilize different evaluation criteria in the future and expand CBC3D’s capability to provide accurate geometric and topologic representation for a more robust range of test cases.

In its pre-processing stage, CBC3D implements a relabeling algorithm that processes disconnected regions in the image as different materials (section 3.2.2), leading to a more localized refinement per material. A more efficient approach to handling non-manifold voxel connectivity would be to solve the topological issues in the meshing stage (i.e., ensuring that the generated mesh does not contain any non-manifold edges or vertices). Since the red-green templates utilized by CBC3D’s adaptive refinement do not guarantee a manifold mesh, this problem requires further consideration and will be addressed in the future. It should also be noted that CBC3D’s output is non-deterministic due to: (i) the randomness introduced in the relabeling operations used for eliminating non-manifold voxel connectivity (section 3.2.3) and (ii) the parallelized computations of EDTs, source points, and target points. While this may be an issue for certain applications, the targeted biomedical simulations do not require deterministic output from the mesher. Instead, they focus on the aforementioned requirements - fidelity, quality, real-time performance, and smoothness.

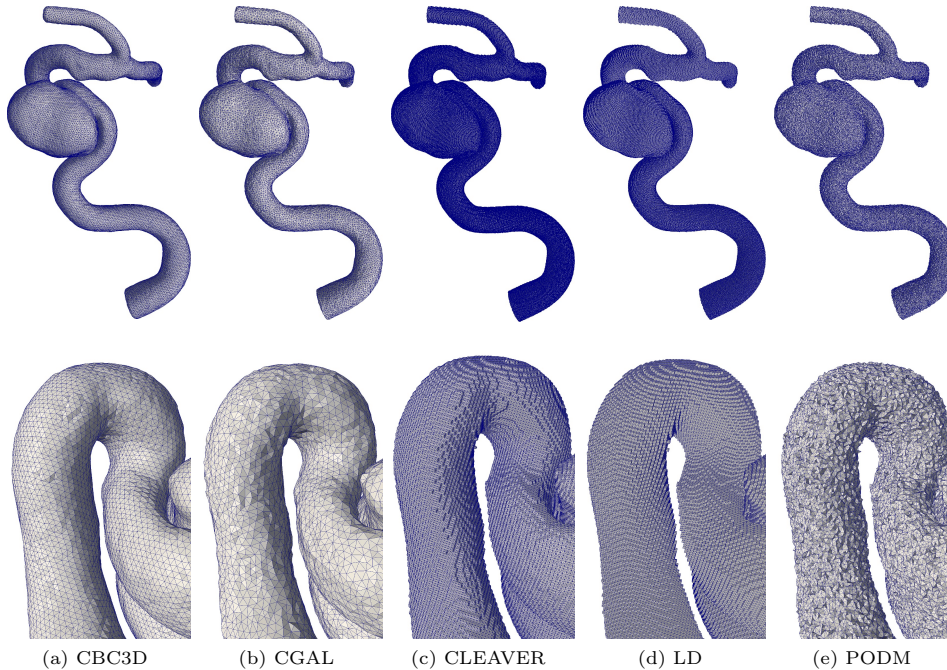
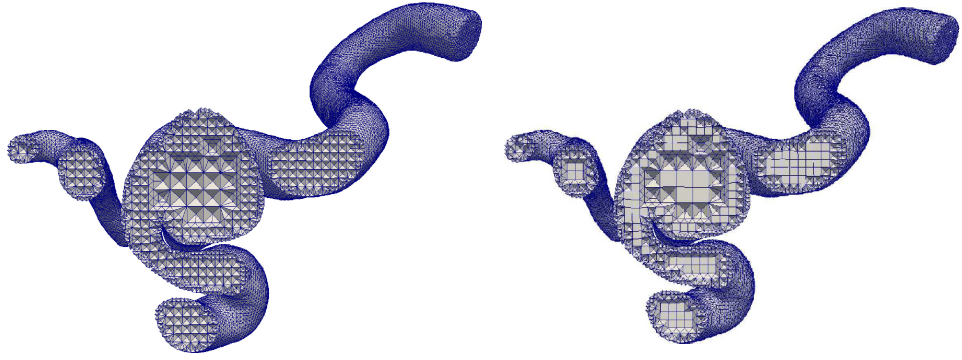


Fig. 24 Comparison between the surface meshes of case 1 (Cavernous Aneurysm). Each column corresponds to a single method. The bottom row depicts a closer view of the surface. Among the methods, only CBC3D approximates the voxelized segmentation with a smooth surface that reflects a certain degree of visual reality.

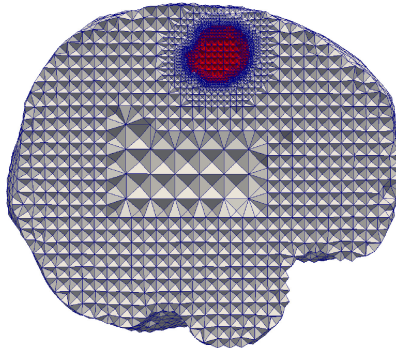
Currently, the smoothness of the surface meshes generated from data with small features (e.g., Micro-CT imaging of stents) is not satisfactory (Figure 17a). The smoothness can be improved by incorporating a semi-analytic technique and by using a priori knowledge of the radii of the stent wires. Also, the mechanical properties (i.e., Young’s modulus, Poisson ratio) of the materials adjacent to the mesh interfaces can be appropriately adjusted to improve smoothness. The accuracy of CFD simulations used to study localized alterations in hemodynamics due to different stent devices may be improved further with boundary layer meshes [62] and anisotropic meshes [18].

In table 7, PODM exhibits better performance than CBC3D (leading to its integration within a non-rigid registration method for image-guided neurosurgery [59]); however, CBC3D is proven to generate meshes of better quality than PODM [10] (CBC3D also exhibits a runtime of approximately 5-15 minutes when meshing smaller geometries [12]). While CBC3D parallelizes the calculation of EDTs during adaptive refinement and computes the source and target points in parallel during deformation (achieving better runtime performance compared to its earlier versions [11, 12]), more work must be done to parallelize the method fully. A Structured Adaptive Mesh Refinement (SAMR) scheme [63] can improve the performance and the gradation of the sequential CBC3D method. SAMR initially discretizes the image domain with a uniform mesh, and then generates finer sub-meshes (components) near the areas where

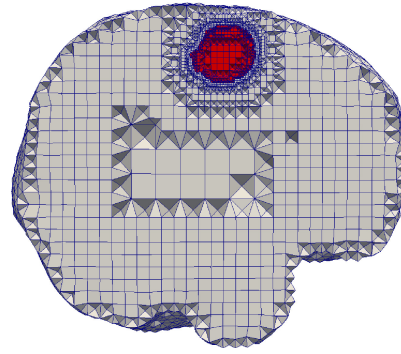


(a) $\#(\text{pts, tets})$: (63.6, 272.2) K; α_{min} : 5.07°

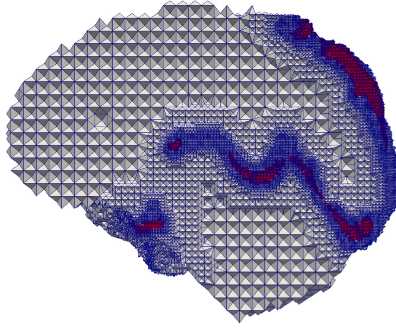
(b) $\#(\text{pts, tets, pyrs, hexs})$: (62.1, 246.7, 4.0, 1.4) K; α_{min} : 5.31° ; J_{min} : 0.47



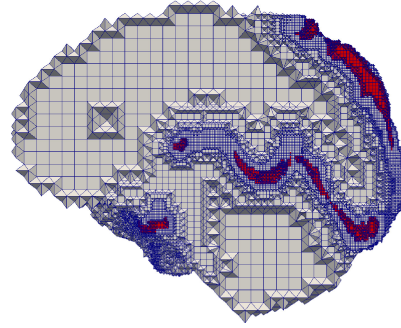
(c) $\#(\text{pts, tets})$: (101.4, 578.3) K; α_{min} : 8.89°



(d) $\#(\text{pts, tets, pyrs, hexs})$: (78.8, 226.8, 40.2, 22.5) K; α_{min} : 5.19° ; J_{min} : 0.55



(e) $\#(\text{pts, tets})$: (887.0, 5059.9) K; α_{min} : 29.94°



(f) $\#(\text{pts, tets, pyrs, hexs})$: (679.4, 1839.1, 364.9, 207.5) K; α_{min} : 29.18° ; J_{min} : 0.98

Fig. 25 Comparison between adaptive tetrahedral meshes and their corresponding mixed meshes all generated with CBC3D. The top, middle, and bottom row correspond to cases 1, 2, and 3, respectively. $\alpha_{min} \in [0, 180]$: minimum dihedral angle (for tetrahedra); $J_{min} \in [0, 1]$: minimum scaled Jacobian (for pyramids and hexahedra).

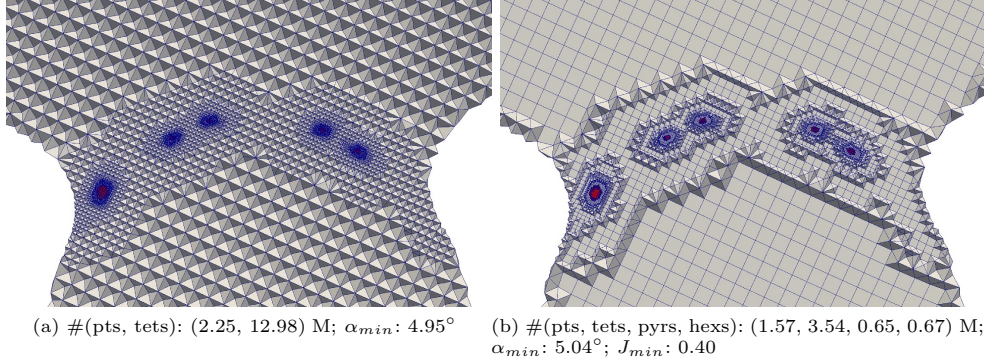


Fig. 26 Comparison between adaptive tetrahedral mesh and its corresponding mixed mesh for case 4 (Lumen-LVIS Stent). The meshes were generated with CBC3D. $\alpha_{\min} \in [0, 180]$: minimum dihedral angle (for tetrahedra); $J_{\min} \in [0, 1]$: minimum scaled Jacobian (for pyramids and hexahedra).

the physics are “changing.” SAMR uses an independent Partial Differential Equation (PDE) solver in each mesh component and assesses the validity of the numerical results. If the numerical solution satisfies the analysis requirements, then the refinement stops; otherwise, SAMR discretizes the mesh with a finer resolution until it obtains an acceptable solution. Each mesh component has its own solution vector, which is computed independently from the solution of the other mesh components. This is important for the parallelization of the SAMR scheme, where each mesh component is associated with a single core/node, and the under-utilized cores/nodes (e.g., those that compute the PDE solution in a smaller number of cells or mesh points) can automatically request additional work from the remaining cores/nodes.

A previous multi-tissue version of CBC3D [5] is available as a stand-alone ITK library and is integrated [64] within an interactive simulator for neurosurgical procedures involving brain Arteriovenous Malformations (AVM) developed in SOFA, a framework for real-time medical simulations [65]. A single-material version is available within the 3D Slicer package for visualization and image analysis. As mentioned previously, the current version of the code utilizes shared-memory processing cores during the smoothing component of its algorithm – this is outside the scope of this paper. For more information on integration and parallelization, see [66].

6 Conclusion

An adaptive, image-to-mesh conversion method called CBC3D is presented. This method builds upon previous work [5, 11, 12], maintaining the ability to generate meshes of good quality that were shown to enhance non-rigid registration solver performance and reduce error when compared to other image-to-mesh conversion methods [10]. CBC3D initially discretizes a segmented, labeled image with a uniform BCC lattice of high-quality tetrahedra. It then employs red-green templates to subdivide the lattice near the segmented boundaries while ensuring mesh conformity (i.e., manifold

connectivity between mesh regions representing different tissues). Finally, the generated surfaces are deformed to their corresponding tissue boundaries to improve fidelity while maintaining quality.

CBC3D offers several attributes : (i) it provides accurate geometric and topologic representation for the cases presented in our evaluation, (ii) it provides material-dependent mesh resolution to reduce element count, (iii) it maintains good element quality during mesh deformation, (iv) it reduces subsequent memory and CPU requirements for the solver by introducing mixed elements and further reduces the size of the mesh, and (v) it improves the overall reliability and portability of the code as it builds upon the ITK open-source, cross-platform system. The CBC3D software is available as (i) a stand-alone ITK library (previous multi-tissue version based on [5]), (ii) a 3D Slicer¹⁰ extension (single-tissue version), and (iii) a SOFA plugin within an interactive simulator (multi-tissue version with limited features) for neurosurgical procedures involving brain Arteriovenous Malformations (AVM).

The present method is compared with four image-to-mesh conversion codes commonly used in industry and academia: CGAL’s 3D Mesh Engine (v4.5.2), CLEAVER [39] (v1.5.4), Lattice-Derefinement (LD) [44], and PODM [30]. Previous work on lattice-based meshing [44] show that quality, fidelity, and size criteria conflict with one another, as it is challenging to balance all three satisfactorily. Our evaluation results indicate that the CBC3D meshes (i) exhibit high fidelity, (ii) keep the element count reasonably low, and (iii) exhibit good element quality.

7 Acknowledgments

This work is in part funded by NSF grant no. CCF-1439079, the NASA Transformational Tools and Technologies Project (NNX15AU39A) of the Transformative Aeronautics Concepts Program under the Aeronautics Research Mission Directorate, Richard T. Cheng Endowment, the Modeling and Simulation fellowship of Old Dominion University, and the National Institute of General Medical Sciences of the National Institutes of Health under Award Number 1T32GM140911-03. This paper describes objective technical results and analysis. Any subjective views or opinions expressed in the paper do not necessarily represent the views of the National Institutes of Health or the United States Government.

References

- [1] Archip, N., Fedorov, A., Lloyd, B., Chrisochoides, N., Golby, A., Black, P.M., Warfield, S.K.: Integration of patient specific modeling and advanced image processing techniques for image-guided neurosurgery. In: Cleary, K.R., Galloway, J. Robert L. (eds.) *Medical Imaging 2006: Visualization, Image-Guided Procedures, and Display*. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 6141, pp. 422–429 (2006). <https://doi.org/10.1117/12.653930>

¹⁰<https://www.slicer.org>

- [2] Talos, I., Archip, N.: Volumetric non-rigid registration for MRI-guided brain tumor surgery. Technical report, Surgical Planning Laboratory, Department of Radiology, Brigham and Women’s Hospital, Harvard Medical School (2007)
- [3] Archip, N., Clatz, O., Whalen, S., Kacher, D., Fedorov, A., Kot, A., Chrisochoides, N., Jolesz, F., Golby, A., Black, P.M., Warfield, S.K.: Non-rigid alignment of pre-operative MRI, fMRI, and DT-MRI with intra-operative MRI for enhanced visualization and navigation in image-guided neurosurgery. *NeuroImage* **35**(2), 609–624 (2007) <https://doi.org/10.1016/j.neuroimage.2006.11.060>
- [4] Drakopoulos, F., Liu, Y., Foteinos, P., Chrisochoides, N.P.: Towards a real time multi-tissue adaptive physics based non-rigid registration framework for brain tumor resection. *Frontiers in Neuroinformatics* **8**(11) (2014) <https://doi.org/10.3389/fninf.2014.00011>
- [5] Drakopoulos, F., Ortiz, R., Enquobahrie, A., Sasaki-Adams, D., Chrisochoides, N.: Tetrahedral image-to-mesh conversion software for anatomic modeling of arteriovenous malformations. *Procedia Engineering* **124**, 278–290 (2015) <https://doi.org/10.1016/j.proeng.2015.10.139> . 24th International Meshing Roundtable
- [6] Dholakia, R.J., Drakopoulos, F., Sadasivan, C., Jiao, X., Fiorella, D.J., Woo, H.H., Lieber, B.B., Chrisochoides, N.: High fidelity image-to-mesh conversion for brain aneurysm/stent geometries. In: *IEEE International Symposium on Biomedical Imaging: From Nano to Macro* (2015)
- [7] Dholakia, R.J., Pagano, A., Drakopoulos, F., Kappel, A., Sadasivan, C., Jiao, X., Fiorella, D.J., Chrisochoides, N., Woo, H.H., Lieber, B.B.: In vitro and computational fluid dynamics comparison of the flow diversion efficacy of five commercial stents. In: *Summer Biomechanics, Bioengineering and Biotransport Conference* (2015)
- [8] Kazakidi, A., Drakopoulos, F., Sadasivan, C., Chrisochoides, N., Ekaterinaris, J., Lieber, B.B.: Simulation of blood flow diversion in cerebral aneurysms. In: *European Congress on Computational Methods in Applied Sciences and Engineering* (2016)
- [9] Babuška, I., Aziz, A.K.: On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis* **13**(2), 214–226 (1976)
- [10] Foteinos, P.A., Liu, Y., Chernikov, A.N., Chrisochoides, N.P.: An evaluation of tetrahedral mesh generation for nonrigid registration of brain mri, 131–142 (2011)
- [11] Fedorov, A., Chrisochoides, N., Kikinis, R., Warfield, S.: Tetrahedral mesh generation for medical imaging. In: *8th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2005)* (2005)
- [12] Liu, Y., Foteinos, P., Chernikov, A., Chrisochoides, N.: Mesh deformation-based

- multi-tissue mesh generation for brain images. *Engineering with Computers* **28**(4), 305–318 (2012) <https://doi.org/10.1007/s00366-012-0265-y>
- [13] Xu, J., Chernikov, A.N.: Curvilinear triangular discretization of biomedical images. In: Harrison, R., Li, Y., Măndoiu, I. (eds.) *Bioinformatics Research and Applications*, pp. 343–354. Springer, Cham (2015)
- [14] Xu, J., Chernikov, A.N.: Automatic curvilinear quality mesh generation driven by smooth boundary and guaranteed fidelity. *Procedia Engineering* **82**, 200–212 (2014) <https://doi.org/10.1016/j.proeng.2014.10.384> . 23rd International Meshing Roundtable (IMR23)
- [15] Zhang, Y.: *Geometric Modeling and Mesh Generation from Scanned Images*. Chapman & Hall, Boca Raton, FL (2016)
- [16] Clerici, F., Ferro, N., Marconi, S., Micheletti, S., Negrello, E., Perotto, S.: Anisotropic adapted meshes for image segmentation: Application to three-dimensional medical data. *SIAM Journal on Imaging Sciences* **13**(4), 2189–2212 (2020) <https://doi.org/10.1137/20M1348303> <https://doi.org/10.1137/20M1348303>
- [17] Carson, J.P., Kuprat, A.P., Jiao, X., Dyedov, V., Pin, F.d., Guccione, J.M., Ratcliffe, M.B., Einstein, D.R.: Adaptive generation of multimaterial grids from imaging data for biomedical lagrangian fluid–structure simulations. *Biomechanics and Modeling in Mechanobiology* **9**, 187–201 (2010) <https://doi.org/10.1007/s10237-009-0170-5>
- [18] Zhang, H., Jiao, Y., Johnson, E., Zhan, L., Zhang, Y., Shimada, K.: Modelling anisotropic material property of cerebral aneurysms for fluid–structure interaction simulation. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* **1**(3), 164–174 (2013) <https://doi.org/10.1080/21681163.2013.776270> <https://doi.org/10.1080/21681163.2013.776270>
- [19] Joshi, B., Fedorov, A., Chrisochoides, N., Warfield, S.K., Ourselin, S.: A quantitative assessment of approaches to mesh generation for surgical simulation. *Eng. with Comput.* **24**(4), 417–430 (2008)
- [20] Fedorov, A., Chrisochoides, N.: Tetrahedral mesh generation for non-rigid registration of brain mri: Analysis of the requirements and evaluation of solutions. In: Garimella, R.V. (ed.) *Proceedings of the 17th International Meshing Roundtable*, pp. 55–72. Springer, Berlin, Heidelberg (2008)
- [21] Ito, Y., Shih, A.M., Erukala, A.K., Soni, B.K., Chernikov, A., Chrisochoides, N.P., Nakahashi, K.: Parallel unstructured mesh generation by an advancing front method. *Mathematics and Computers in Simulation* **75**(5), 200–209 (2007) <https://doi.org/10.1016/j.matcom.2006.12.008> . *Applied Scientific Computing: Advanced Grid Generation, Approximation and Simulation*

- [22] Rivara, M.-C., Chrisochoides, N.: Parallel refinement of tetrahedral meshes using terminal-edge bisection algorithm. In: International Meshing Roundtable, pp. 427–436 (2004)
- [23] Rivara, M.-C., Calderon, C., Fedorov, A., Chrisochoides, N.: Parallel decoupled terminal-edge bisection method for 3d mesh generation. *Engineering with Computers* **22**, 111–119 (2006) <https://doi.org/10.1007/s00366-006-0013-2>
- [24] Boltcheva, D., Yvinec, M., Boissonnat, J.-D.: Mesh generation from 3D multi-material images. In: Yang, G.-Z., Hawkes, D., Rueckert, D., Noble, A., Taylor, C. (eds.) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009*. Lecture Notes in Computer Science, vol. 5762, pp. 283–290. Springer, London, UK (2009)
- [25] Pons, J.-P., Ségonne, F., Boissonnat, J.-D., Rineau, L., Yvinec, M., Keriven, R.: High-quality consistent meshing of multi-label datasets. In: *Biennial International Conference on Information Processing in Medical Imaging*, pp. 198–210 (2007). Springer
- [26] Fuchs, M., Wagner, M., Kastner, J.: Boundary element method volume conductor models for eeg source reconstruction. *Clinical Neurophysiology* **112**(8), 1400–1407 (2001)
- [27] Shewchuk, J.R.: Tetrahedral mesh generation by Delaunay refinement. In: *Proceedings of the Fourteenth Annual Symposium on Computational Geometry*. SCG '98, pp. 86–95. ACM, New York, NY, USA (1998)
- [28] Rineau, L., Yvinec, M.: A generic software design for Delaunay refinement meshing. *Computational Geometry* **38**(1-2), 100–110 (2007)
- [29] Cheng, S.-W., Dey, T.K., Edelsbrunner, H., Facello, M.A., Teng, S.-H.: Silver exudation. *Journal of the ACM (JACM)* **47**(5), 883–904 (2000)
- [30] Foteinos, P.A., Chrisochoides, N.P.: High quality real-time image-to-mesh conversion for finite element simulations. *Journal of Parallel and Distributed Computing* **74**(2), 2123–2140 (2014)
- [31] Feng, D., Tzolakis, C., Chernikov, A.N., Chrisochoides, N.: Scalable 3d hybrid parallel delaunay image-to-mesh conversion algorithm for distributed shared memory architectures. *Comput. Aided Des.* **85**, 10–19 (2017)
- [32] Foteinos, P., Chrisochoides, N.: 4d space-time delaunay meshing for medical images. In: Sarrate, J., Staten, M. (eds.) *Proceedings of the 22nd International Meshing Roundtable*, pp. 223–240. Springer, Cham (2014)
- [33] Foteinos, P., Chernikov, A., Chrisochoides, N.: Guaranteed quality tetrahedral delaunay meshing for medical images. In: *7th International Symposium on*

Voronoi Diagrams in Science and Engineering, Laval University, Quebec City, Canada, pp. 215–223 (2010)

- [34] Foteinos, P., Chrisochoides, N.: High-quality multi-tissue mesh generation for finite element analysis. In: MeshMed, Workshop on Mesh Processing in Medical Image Analysis (MICCAI), pp. 18–28 (2011)
- [35] Foteinos, P., Chrisochoides, N.: High-quality multi-tissue mesh generation for finite element analysis. Image-Based Geometric Modeling and Mesh Generation, 161–172 (2012). Editor Yongjie (Jessica) Zhang
- [36] Foteinos, P., Chrisochoides, N.: Dynamic parallel 3d delaunay triangulation. In: International Meshing Roundtable, pp. 3–20. Springer, Paris, France (2012)
- [37] Foteinos, P., Chrisochoides, N.: High quality real-time image-to-mesh conversion for finite element simulations. In: Proceedings of the 27th International ACM Conference on International Conference on Supercomputing. ICS '13, pp. 233–242. ACM, New York, NY, USA (2013). <https://doi.org/10.1145/2464996.2465439>
- [38] Chew, L.P.: Guaranteed-quality Delaunay meshing in 3D (short version). In: Proceedings of the Thirteenth Annual Symposium on Computational Geometry. SCG '97, pp. 391–393. ACM, New York, NY, USA (1997)
- [39] Bronson, J., Levine, J.A., Whitaker, R.: Lattice cleaving: A multimaterial tetrahedral meshing algorithm with guarantees. Visualization and Computer Graphics, IEEE Transactions on **20**(2), 223–237 (2014) <https://doi.org/10.1109/TVCG.2013.115>
- [40] Zhang, Y., Hughes, T.J.R., Bajaj, C.L.: An automatic 3d mesh generation method for domains with multiple materials. Computer Methods in Applied Mechanics and Engineering **199**(5), 405–415 (2010) <https://doi.org/10.1016/j.cma.2009.06.007> . Computational Geometry and Analysis
- [41] Zhang, Y., Bajaj, C., Sohn, B.-S.: 3d finite element meshing from imaging data. Computer Methods in Applied Mechanics and Engineering **194**(48), 5083–5106 (2005) <https://doi.org/10.1016/j.cma.2004.11.026>
- [42] Labelle, F., Shewchuk, J.R.: Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. In: ACM Transactions on Graphics (TOG), vol. 26, p. 57 (2007). ACM
- [43] Zhang, Y., Bajaj, C., Sohn, B.-S.: 3D finite element meshing from imaging data. Computer Methods in Applied Mechanics and Engineering **194**(48), 5083–5106 (2005)
- [44] Chernikov, A.N., Chrisochoides, N.P.: Multitissue tetrahedral image-to-mesh conversion with guaranteed quality and fidelity. SIAM Journal on Scientific

Computing **33**(6), 3491–3508 (2011)

- [45] Zhang, Y.: Challenges and advances in image-based geometric modeling and mesh generation. In: *Image-Based Geometric Modeling and Mesh Generation. Lecture Notes in Computational Vision and Biomechanics*, pp. 1–10. Springer, Dordrecht, Netherlands (2013). https://doi.org/10.1007/978-94-007-4255-0_1
- [46] Zhang, Y., Qian, J.: Resolving topology ambiguity for multiple-material domains. *Computer Methods in Applied Mechanics and Engineering* **247**, 166–178 (2012)
- [47] Liang, X., Zhang, Y.: An octree-based dual contouring method for triangular and tetrahedral mesh generation with guaranteed angle range. *Engineering with Computers* **30**, 211–222 (2014) <https://doi.org/10.1007/s00366-013-0328-8>
- [48] Hu, K., Zhang, Y.J., Xu, G.: Cvt-based 3d image segmentation and quality improvement of tetrahedral/hexahedral meshes using anisotropic gjaquinta-hildebrandt operator. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* **6**(3), 331–342 (2018) <https://doi.org/10.1080/21681163.2016.1244017> <https://doi.org/10.1080/21681163.2016.1244017>
- [49] Liao, T., Zhang, Y., Kekenus-Huskey, P.M., Cheng, Y., Michailova, A., McCulloch, A.D., Holst, M., McCammon, J.A.: Multi-core cpu or gpu-accelerated multiscale modeling for biomolecular complexes. *Computational and Mathematical Biophysics* **1**(2013), 164–179 (2013) <https://doi.org/10.2478/mlbmb-2013-0009>
- [50] Radovitzky, R., Ortiz, M.: Tetrahedral mesh generation based on node insertion in crystal lattice arrangements and advancing-front-Delaunay triangulation. *Computer Methods in Applied Mechanics and Engineering* **187**(3-4), 543–569 (2000)
- [51] Molino, N., Bridson, R., Teran, J., Fedkiw, R.: A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In: *IMR*, pp. 103–114 (2003)
- [52] Freitag, L., Knupp, P., Munson, T., Shontz, S.: A comparison of optimization software for mesh shape-quality improvement problems. Technical report, Argonne National Lab., IL (US) (2002)
- [53] Maurer, C.R., Qi, R., Raghavan, V.: A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(2), 265–270 (2003)
- [54] Antiga, L., Piccinelli, M., Botti, L., Ene-Iordache, B., Remuzzi, A., Steinman, D.: An image-based modeling framework for patient-specific computational hemodynamics. *Medical & Biological Engineering & Computing* **46**(11), 1097–1112 (2008)

- [55] Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3D surface construction algorithm. In: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '87, pp. 163–169. ACM, New York, NY, USA (1987)
- [56] Bathe, K.-J.: Finite Element Procedures. Prentice-Hall, Inc., New Jersey, USA (2006)
- [57] Kincaid, D.R., Respass, J.R., Young, D.M., Grimes, R.R.: Algorithm 586: ITPACK 2C: A FORTRAN package for solving large sparse linear systems by adaptive accelerated iterative methods. ACM Transactions on Mathematical Software (TOMS) **8**(3), 302–322 (1982)
- [58] Knupp, P.M.: Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I-A framework for surface mesh optimization. International Journal for Numerical Methods in Engineering **48**(3), 401–420 (2000)
- [59] Drakopoulos, F., Tsolakis, C., Angelopoulos, A., Liu, Y., Yao, C., Kavazidi, K.R., Foroglou, N., Fedorov, A., Frisken, S., Kikinis, R., Golby, A., Chrisochoides, N.: Adaptive physics-based non-rigid registration for immersive image-guided neuronavigation systems. Frontiers in Digital Health **2** (2021) <https://doi.org/10.3389/fdgth.2020.613608>
- [60] Angelopoulos, A., Chrisochoides, N.: Deep learning real-time adaptive physics-based non-rigid registration for accurate geometry representation of brain in modeling deformation during glioma resection. In: Old Dominion University Undergraduate Research Symposium (2019)
- [61] Commandeur, F., Velut, J., Acosta, O.: A VTK algorithm for the computation of the hausdorff distance. The VTK Journal (2011)
- [62] Dyedov, V., Einstein, D.R., Jiao, X., Kuprat, A.P., Carson, J.P., Pin, F.: Variational generation of prismatic boundary-layer meshes for biomedical computing. International Journal for Numerical Methods in Engineering **79**(8), 907–945 (2009) <https://doi.org/10.1002/nme.2583>
- [63] Baden, S.B., Chrisochoides, N.P., Gannon, D.B., Norman, M.L.: Structured Adaptive Mesh Refinement (SAMR) Grid Methods. Springer, New York, USA (2000)
- [64] Drakopoulos, F., Chrisochoides, N.: Tetrahedral image-to-mesh conversion for anatomical modeling and surgical simulations. In: VMASC 2015 Capstone Conference (2015)
- [65] Faure, F., Duriez, C., Delingette, H., Allard, J., Gilles, B., Marchesseau, S., Talbot, H., Courtecuisse, H., Bousquet, G., Peterlik, I., Cotin, S.: In: Payan, Y. (ed.)

SOFA: A Multi-Model Framework for Interactive Physical Simulation, pp. 283–321. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/8415_2012_125 . http://dx.doi.org/10.1007/8415_2012_125

- [66] Drakopoulos, F.: Finite element modeling driven by health care and aerospace applications. PhD thesis, Old Dominion University (2017)