Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 00 (2014) 000–000

Procedia Engineering

www.elsevier.com/locate/procedia

23rd International Meshing Roundtable (IMR23)

Tetrahedral Image-To-Mesh Conversion Software for Anatomic Modeling of Arteriovenous Malformations

Fotis Drakopoulos[a], Ricardo Ortiz[b], Andinet Enquobahrie[b], Deanna Sasaki-Adams[c], Nikos Chrisochoides[a,*]

[a]CRTC Lab, Department of Computer Science, Old Dominion University, Norfolk, VA 23529, USA
[b]Kitware, Inc., Carrboro, NC, 27510, USA
[c]Department of Neurosurgery, University of North Carolina at Chapel Hill, Chapel Hill, NC, 27599, USA

Abstract

We describe a new implementation of an adaptive multi-tissue tetrahedral mesh generator targeting anatomic modeling of Arteriovenous Malformation (AVM) for surgical simulations. Our method, initially constructs an adaptive Body-Centered Cubic (BCC) mesh of high quality elements. Then, it deforms the mesh surfaces to their corresponding physical image boundaries, hence, improving the mesh fidelity and smoothness. Our deformation scheme, which builds upon the ITK toolkit, is based on the concept of energy minimization, and relies on a multi-material point-based registration. It uses non-connectivity patterns to implicitly control the number of the extracted feature points needed for the registration, and thus, adjusts the trade-off between the achieved mesh fidelity and the deformation speed. While many medical imaging applications require robust mesh generation, there are few codes available to the public. We compare our implementation with two similar open-source image-to-mesh conversion codes: (1) Cleaver from US, and (2) CGAL from EU. Our evaluation is based on five isotropic/anisotropic segmented images, and relies on metrics like geometric & topologic fidelity, mesh quality, gradation and smoothness. The implementation we describe is open-source and it will be available within: (i) the 3D Slicer package for visualization and image analysis from Harvard Medical School, and (ii) an interactive simulator for neurosurgical procedures involving vasculature using SOFA, a framework for real-time medical simulation developed by INRIA.

Keywords: Mesh generation, FEM, Simulation, SOFA, 3D Slicer

1. Intoduction

Image-to-Mesh (I2M) 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 [1–3], interactive surgery simulation for training young clinicians, and endovascular flow diversion for aneurysm geometries [4]. The FE method is essential in modeling tissue deformation for these applications. An intrinsic

[*] Corresponding author. Tel.: +1 (757) 683-77150
E-mail address: nikos@cs.odu.edu

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 parameterized, i.e., it is defined by means of constructive solid geometry primitives, or explicitly defined, e.g., through the boundary discretization, as a collection of patches. In order to convert the isosurface data into a tetrahedral mesh, one can either (1) recover the parametrized object surface and follow up with a conventional mesh generation technique, or (2) use a mesh generation method, which operates directly on this type of data. In this paper we present an open-source implementation of an adaptive multi-tissue tetrahedral mesh generator (CBC3D). The CBC3D operates directly on segmented multi-labeled image data pertinent to Arteriovenous Malformation (AVM), i.e., tangled bundles of abnormal and arbitrary thin blood vessels connecting arteries and veins in the brain. Specifically, the CBC3D meshing method targets the anatomical modeling of AVM for surgical simulations.

The I2M software should meet the requirements of three important components for AVM simulations: (1) level-of-detail of tissue morphology (to satisfy the computational complexity of the collision detection module responsible to managing correctly intra-surgical motion), (2) accuracy of the representation of the geometries (to capture faithful response of the biomechanical properties of the tissues involved in the simulation), and (3) visual reality (to meet the clinical requirements of the surgical simulation).

Medical surgical simulations are classified into two main categories. Predictive simulation is part of current and emerging practices in health care that are working towards precision medicine. The goal for predictive simulation is to predict and optimize the outcome of an intervention by using patient-specific, pre-operative image data. Interactive simulation, on the other hand, is part of the effort from medical schools and hospitals to reduce health care costs. Costs are reduced by training surgical residents using virtual environments to achieve certain skill levels without imposing risks on real patients. The input to both categories of surgical simulations is based on patients medical images like MRI, fMRI, CT and micro-CT. The required analysis for the anatomic modeling step of these two categories is slightly different. Predictive simulations require high geometric, topologic and material fidelity at the cost of large-scale mesh generation which likely takes place off-line with respect to the intervention. In contrast, the interactive simulations require real-time mesh generation to meet the time constraints imposed by end-to-end computations in near-haptic rates. These real-time requirements compromise the fidelity of the anatomic modeling. Despite their differences, there are a number of requirements, like element quality, mesh gradation and multi-tissue, multi-resolution meshing, that are common to both types of simulations. In this paper we focus on these common requirements, and, most importantly, we focus on the high geometric, topologic and material fidelity. The real-time constraints imposed by the interactive simulations will be addressed elsewhere in the near future.

In turn this translates to the following requirements for analysis of the mesh generation. The I2M conversion needs to balance trade-offs between several criteria: (i) minimize the mesh size and maximize the element quality by utilizing proper mesh gradation and tissue-specific multi-resolution, (ii) without sacrificing geometric & topologic fidelity, (iii) while maintaining smoothness of the mesh to reflect a certain degree of reality, and (iv) at the same time, allow the upstream numerical computations for collision detection and biomechanical Finite Element Analysis to complete within the time constraints imposed by the surgical simulation.

2. Background

There are various meshing methods that operate on isosurface data. Some methods are based on a Delaunay refinement scheme [5,6]. CGAL, an open-source software library, provides a Delaunay method for meshing segmented multi-labeled images. Our group developed a 3D Delaunay parallel I2M conversion algorithm that recovers the isosurface of the biological object with geometric and topological guarantees, while also providing good shape elements [7]. The problem with the Delaunay refinement is that almost flat tetrahedra, called slivers, can survive. There are a number of optimization techniques to eliminate slivers [5,8,9], and some of them have been shown to produce very good dihedral angles in practice.

A large number of meshing methods are based on lattice space-tree (e.g., octree in 3D) decomposition, thus providing adaptively sized, high quality elements for a number of medical applications [10,11]. Usually, a regular background mesh tessellates the image and the surface is recovered by finding the points of intersection of the mesh with the surface of the object. Then the mesh vertices are warped onto the model surface, or new vertices are inserted on the surface, and the mesh is locally modified. The achieved mesh quality depends on thresholds for deciding whether to warp or insert a vertex. In some cases, a mesh decimation step is applied to an octree-based mesh to improve the element shape and/or modify the mesh size [12,13]. In [14] an efficient octree Dual Contouring (DC) method was presented to resolve topology ambiguity for multiple-material domains by analyzing a hybrid octree with both cubic and tetrahedral leaf cells.

Another heuristic approach, generates an adaptive Body-Centered Cubic (BCC) background lattice, and then warps the surfaces of the lattice to the boundaries of the object using either a mass spring system, a finite element constitutive model, or an optimization scheme [15]. [16] creates an adaptive BCC lattice, projects the vertices outside the object to the boundary, and then uses a Delaunay approach to tessellate the object, followed by a smoothing optimization step. The method uses the lattice vertices only for estimating the initial positions of the mesh vertices. In [17], an advancing front technique generates the volumetric mesh using the vertices of a Face-Centered Cubic (FCC) lattice.

Our previous work [18,19] extends [15] to fairly regular multi-material geometries, and employs a physics-based mesh deformation scheme to warp the surfaces of a BCC mesh to the boundaries of the object. The deformation scheme takes fidelity, smoothing, and quality into account. The advantage of using a physically based deformation scheme is that it can forecast how the mesh is likely to respond to the deformations it will experience during a simulation.

In this paper, we present additional improvements for the reliable and faithful representation of multiple materials and their corresponding resolutions, i.e., CBC3D is a multi-tissue and multi-resolution mesh generator of fairly complex geometries. We introduce a new implementation of the mesh deformation scheme presented in [19], which builds upon the ITK toolkit and improves the overall robustness and portability of the CBC3D method. The new approach relies on a multi-material point-based registration, and uses non-connectivity patterns to implicitly control the number of the extracted feature points needed for the registration, and to adjust the trade-off between the achieved mesh fidelity and the deformation speed. We evaluated and compared our implementation with two other open-source meshing codes: CGAL v4.5.2, and Cleaver v1.5.4. Our qualitative/quantitative evaluation is based on isotropic/anisotropic segmented multi-labeled pre-operative MR images, including a tumor, a nidus, a pair of lateral ventricles, and an AVM. In this evaluation, we exclude any non publicly available meshing methods developed by our group [7,12]. The CBC3D software is open-source and available within: (i) the 3D Slicer package for visualization and image analysis (single-tissue version; this multi-tissue code will become available before the IMR meeting), and (ii) an interactive simulator for neurosurgical procedures involving brain AVM, developed in SOFA, a framework for real-time medical simulation (multi-tissue version).

3. Method

Our approach uses a segmented multi-labeled image as input, and produces an adaptive tetrahedral mesh. Subsection 3.1 describes the segmentation algorithm used to obtain the AVM labeled image. Subsections 3.2, 3.3 describe the steps of the implementation: (1) BCC mesh generation and refinement, and (2) mesh deformation, respectively.

3.1. AVM Segmentation

Before segmenting the vessels from the medical images, we apply a skull stripping algorithm that removes the skull bone from the image. We employ the Brain Extraction Tool (BET) for this purpose [20]. The level set segmentation algorithm for the vessels structures uses the Vascular Modeling Toolkit (VMTK) [21]. 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. In order to extract the surface from the image we run the output image through Marching Cubes [22] with level zero. We use a fast marching initialization, 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 contain labels each one representing a single tissue. These labeled images are then used as input to the meshing algorithm.

3.2. BCC Mesh Generation and Refinement

Image pre-processing. Our previous work [18,19], re-samples the input image to an isotropic unit spacing image, to avoid the transformation of index to physical coordinates, and thus, to make the implementation simpler and faster. Although this approach is sufficient for fairly regular geometries, in the case of complex data (e.g. AVM), the image down-sampling can cause an invalid image topology. Figure 1 illustrates such an example. On the contrary, the CBC3D operates on either isotropic or anisotropic data, and optionally performs only global up-sampling.

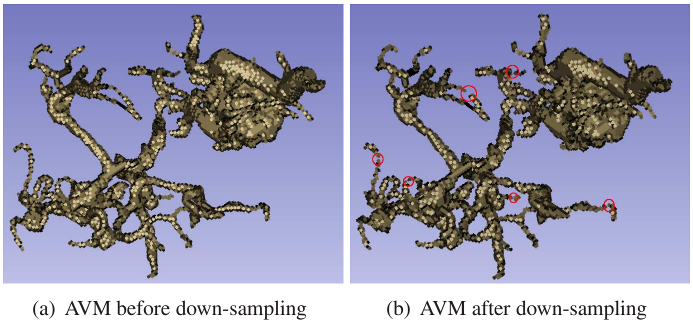

(a) AVM before down-sampling (b) AVM after down-sampling

Fig. 1. Brain Arteriovenous Malformation (AVM) segmentation, before and after down-sampling. The red circles indicate locations of disconnectivity or a non-manifold image topology. The image spacing (mm) before and after down-sampling is [0.879, 0.879, 0.879] and [1, 1, 1], respectively. The image intensity at non-integer pixel positions is computed with a nearest neighbor interpolator.

BCC lattice construction. The bounding box of the input object is discretized with a regular BCC lattice. The distance between the vertices of the lattice can be adjusted with the parameter *BCC_size* (Table 3). Next, the tetrahedra located completely outside of the object are discarded. The resulting elements are of the best quality possible with the regular space tiling [16].

Adaptive refinement. The lattice is refined using a red (regular) - green (irregular) subdivision (Figure 2). We refine only those elements that cross the object surface or the tissue interfaces. Initially, all lattice tetrahedra are labeled with a red color. A red tetrahedron can be subdivided into eight children (1:8 regular refinement), and each child is labeled with a red color, as shown in Figure 2(a). There are three choices for the internal edge of the tetrahedron. If the shortest one is selected, the resulting eight child tetrahedra are exactly the same with the parent tetrahedron 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, including the three cases depicted in Figures 2(b)-2(d), is performed. To quantitatively evaluate the similarity between the sub-mesh and the tissue region, we define S_1 as the set of all voxels in the sub-mesh, S_2 as the set of all voxels in the tissue region, and $S_1 \cap S_2$ as the point-set shared by the sub-mesh and the tissue 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 tissue region. The refinement criterion is defined as follows: Refine the sub-mesh if $F > F_1$ or $F > F_2$, where $F \in (0, 1]$ is the desired fidelity (Table 3). F_1 and F_2 rely on the number of voxels inside a tetrahedron. In the case where no voxels can identify inside an element, we locally up-sample the region defined by the tetrahedron's bounding box.

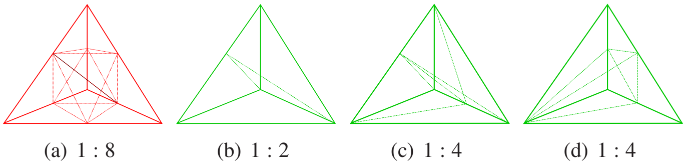

(a) 1 : 8 (b) 1 : 2 (c) 1 : 4 (d) 1 : 4

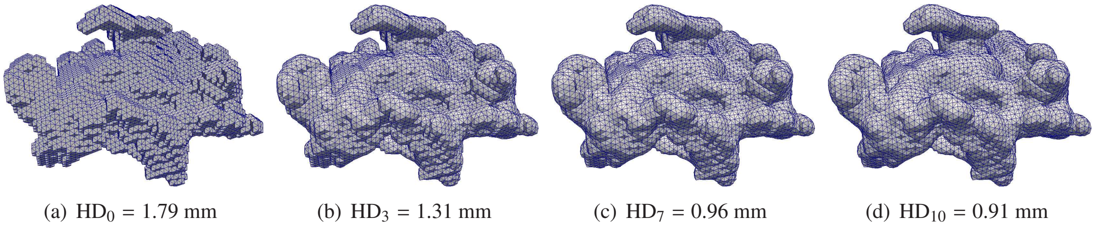
Fig. 2. Red-green refinement. 8, 2, and 4, denote the number of tetrahedrals after the refinement.

Candidate mesh selection. In this step we once again revise the tetrahedral elements which are located close to the surface of the object, and selectively discard those elements which are located completely outside.

3.3. Mesh Deformation

The refinement step guarantees the high quality of the BCC mesh (i.e. worse (min, max) dihedral angles (30°, 116.5°)). However, this mesh does not meet some of the requirements for AVM simulations (i.e. accuracy of the geometric representation, certain degree of visual reality). In this paper, we propose a new implementation of the mesh deformation scheme presented in [19], to deform the mesh surfaces to their corresponding physical image boundaries, and thus, to achieve a higher fidelity and a smoother surface. The mesh deformation performs a number of iterations specified by the user (Table 3). The larger the number of iterations, the higher the fidelity and the smoother the surface.

Point-based registration. The deformation scheme relies on a point-based registration, and builds upon the ITK toolkit, thus, improving the overall reliability and portability of the CBC3D method. The registration is performed between a source and a target point-set. The first set contains the surface/interface vertices of the mesh. The second set contains the surface/interface voxels of the image object. In a typical iteration *i*, for each source point, its potential corresponding target point is located in the neighborhood of the source point. Next, a FE bio-mechanical model is built, the estimated correspondences are applied as constraints to the model, and a linear system of equations is solved to compute the displacements on the mesh vertices. Then, the mesh is updated by adding the computed displacements to the current coordinates of the vertices, and the procedure is repeated until the iteration reaches the maximum number. During the deformation, inverted (flipped) elements might occur, because the magnitude of the correspondence between a source and a target point might be larger than a local element size. The local element size is computed as the average length of the tetrahedral edges adjacent on a source point. In order to reduce the number of possible inverted elements, all estimated correspondences are restricted to be less or equal to a local element size. Besides, after each deformation the tetrahedral elements are checked for inversion. If inverted elements are survived, the mesh deformation stops, and the method recovers the mesh coordinates of the previous iteration. Figure 3 illustrates an example of the mesh deformation step.

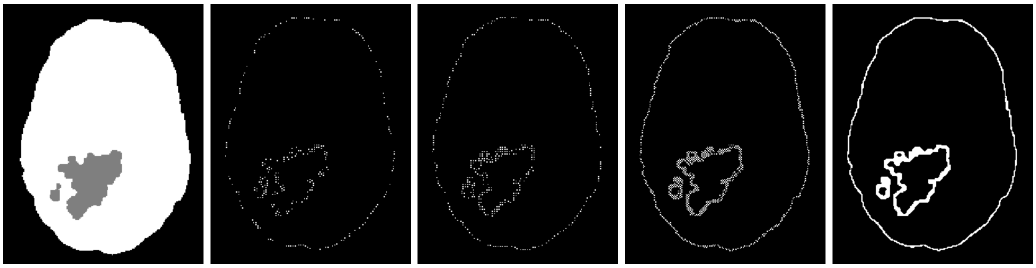


(a) $HD_0 = 1.79$ mm (b) $HD_3 = 1.31$ mm (c) $HD_7 = 0.96$ mm (d) $HD_{10} = 0.91$ mm

Fig. 3. Nidus mesh during a deformation step with 10 iterations. (a) depicts the initial undeformed BCC mesh ($i = 0$). (b), (c), (d), depict the deformed mesh at iterations $i = 3, 7, 10$, respectively. HD_i denotes the mesh fidelity in terms of a Hausdorff Distance metric, at iteration *i*. The smaller the HD, the higher the fidelity. As the number of iterations advances, the mesh exhibits a higher fidelity and a smoother surface.

Adjustable number of target points. Our previous work [19] cannot control the number of the extracted target points needed for the registration. Our new implementation can implicitly control the number of the target points, using non-connectivity patterns. A non-connectivity pattern avoids the selection of neighboring voxels which are connected via a "vertex" (26 non-connectivity), an "edge" (18 non-connectivity), or a "face" (6 non-connectivity), and thus prohibits the extraction of points that are too close to each other. The "no" option disables the non-connectivity patterns, so it maximizes the number of points. Figure 4 depicts the extracted target points for each pattern. Table 1 illustrates the influence of each pattern on the performance of the mesh deformation. The larger the number of the target points, the more accurate the representation of the object's boundaries/interfaces (Figure 4), and thus, the more precise the registration, and the mesh deformation (Table 1). On the other hand, the smaller the number of the target points, the less accurate, but faster the deformation (Table 1). In our experimental evaluation, we use a "face" pattern (Table 3) to balance the deformation accuracy and the speed.

Multi-labeled source/target points. In the registration phase, for each source point, its potential corresponding target point is identified by searching the neighborhood of the source point. Our previous implementation [19], associates each source/target point with a unique image label to denote which tissue it belongs to and, therefore, to restrict the search to the target points which have the same label with the source point. However, in the case of multi-labeled images, a mesh vertex can be adjacent to tetrahedrals of different labels; thus, a unique label is not sufficient to denote which tissues a vertex belongs to. For this reason, the new implementation associates each source/target point with a label-set, instead of a unique label. Specifically, a source point is associated with the labels of the adjacent tetrahedrals on this source point (vertex), and a target point with the labels of the neighboring voxels sharing a face with this target point (voxel). Then, the search for correspondences is restricted between pairs of source and target points with same label-sets.



(a) Segmentation (b) “Vertex” (21510) (c) “Edge” (26387) (d) “Face” (47306) (e) “No” (91906)

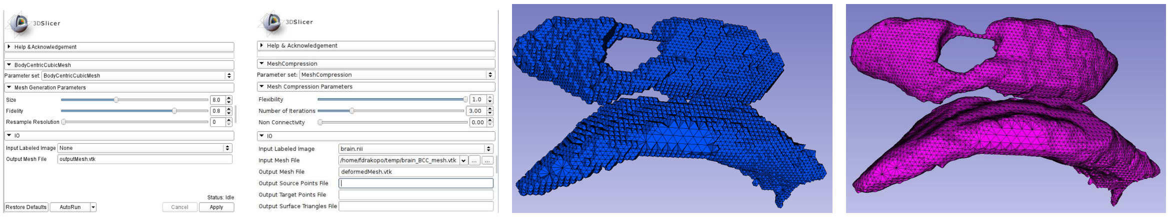
Fig. 4. The extracted target points from a brain-nidus segmentation, depicted for each non-connectivity pattern (“Vertex”, “Edge”, “Face”, and “No”). Each pattern results to a different number/density of points. In the parenthesis we report the number of the points. For simplicity, we depict the points in only one volumetric axial image slice.

Table 1. Mesh deformation performance on a nidus geometry (Figure 3), for each non-connectivity pattern. 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 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. Software Implementation

The CBC3D meshing algorithm is available as: (i) a stand-alone ITK library (multi-tissue version), (ii) a 3D Slicer extension (single-tissue version; the multi-tissue code will become available before the IMR meeting), (iii) a SOFA plugin within an interactive simulator (multi-tissue version).



Parameter set: BodyCentricCubicMesh
Size 8.0
Fidelity 0.8
Resample Resolution 0
Input Labeled Image None
Output Mesh File outputMesh.vtk
Status: Idle

Parameter set: MeshCompression
Flexibility 1.0
Number of Iterations 3.00
Non Connectivity 0.00
Input Labeled Image brain.nii
Input Mesh File /home/fdrakopo/temp/brain_BCC_mesh.vtk
Output Mesh File deformedMesh.vtk
Output Source Points File
Output Target Points File
Output Surface Triangles File

(a) Mesh generation panel (b) Mesh deformation panel (c) Mesh before deformation (d) Mesh after deformation

Fig. 5. CBC3D extension in 3D Slicer. (a), (b) depict the extension’s GUI. (c), (d) depict a ventricle mesh generated with the extension.

4.1. 3D Slicer Extension

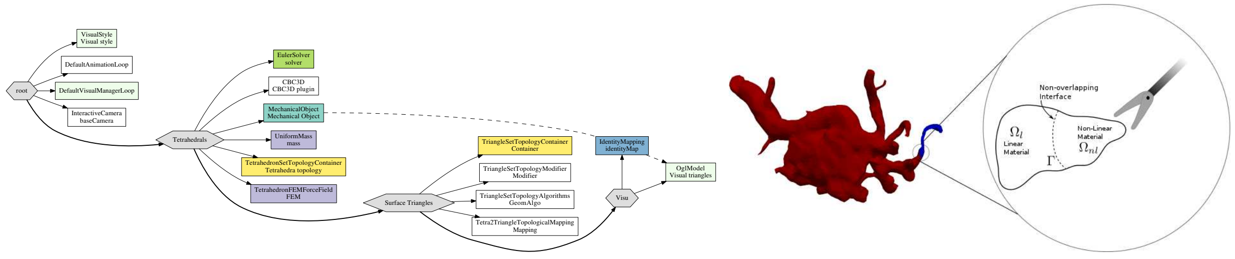
3D Slicer is a free, open-source C++ software application widely used in the medical computing research community. It builds upon VTK and ITK, for visualization and image processing, respectively. Slicer facilitates the clinical translation of medical computing research ideas by providing developers with an easy to use framework for image analysis and visualization. It also provides a template framework that allows for the creation of customized and sophisticated user interfaces. The CBC3D meshing method is available as an extension in 3D Slicer (Figure 5). Slicer supports different types of extensions which can be built outside of the source tree and bundle together one or more modules. The CBC3D extension can be downloaded using Slicer's Extension Manager[1].

4.2. SOFA-based Interactive Simulator

The purpose of the CBC3D image-to-mesh generator is to be used to generate anatomically correct AVM models for surgical simulations. Interactive simulations involving deforming solids are widely used in physically-based simulations. In particular, accurate and efficient simulation tools are critical for the design and development of surgical simulations 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 very difficult problem. Broadly, the main components for most interactive medical simulators require knowledge in the following areas: bio-mechanical and anatomical modeling, collision detection, haptics and visualization. A virtual anatomical model constitute a precise computerized description of a human organ that is commonly generated from medical images, like MRI or CT. In this case it describes an AVM pathology containing draining vessels, feeding arteries and the nidus of vessels bundles as well as surrounding. The brain matter interacts with vessels, fluid and possibly other rigid objects, thus a very important aspect of the simulator is the ability to provide anatomically correct models of the structures involved in the virtual surgical procedure. At the same time, the anatomical model should contain the minimal amount of elements needed to accurately describe the small vessel features.

In this study, we developed an interactive simulator for neurosurgical procedures, involving brain vasculature. The simulator builds upon the SOFA open-source framework for real-time medical simulation. The simulator's development framework consists of FEM biomechanical anatomical modeling and volumetric meshing of vascular structures, coupling collision detection and response with haptic feedback, and GPU-based implementations enabling real-time simulation. It is outside the scope of this paper to extensive talk about the AVM simulator. It will appear elsewhere.

SOFA Plugin. The CBC3D method is also available as a plugin in SOFA. Our simulator uses this plugin to generate a tetrahedral mesh of the AVM pathology. The plugin mechanism exposes SOFA's API and allows us to use all the necessary features and data structures without the need to tightly integrate the CBC3D with the rest of the SOFA library. Figure 6(a) depicts the scene graph of a simple SOFA simulation using the CBC3D plugin to generate a mesh from a segmented image. The visual or collision model is crated by mapping a surface triangle topology on the surface of the tetrahedral mesh.



(a)

(b)

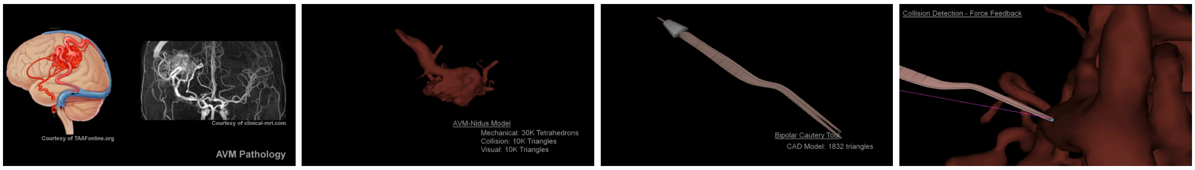
Fig. 6. (a): Scene graph of a sofa simulation using the CBC3D plugin. (b): Conceptual depiction of the hybrid FEM method.

Collision Detection. SOFA's current capabilities in terms of reliable collision detection and response computation are limited. It only performs discrete collision, i.e., it checks for collisions between the mesh at fixed time steps. Furthermore, it has limited capabilities for self-collisions. In our simulator, we employ a continuous collision detection method (Self-CCD) developed in Gamma Lab at the University of North Carolina. This algorithm performs inter and intra objects queries using various culling techniques (i.e. representative triangles [23], connectivity [24] and other non-penetration filters [25]) to speed up computation and robustness.

FEM Simulator. Tissue response is handled by a corotational tetrahedral FEM formulation. Work is undergoing to integrate a hybrid FEM approach to increase performance, depicted in Figure 6(b). The hybrid FEM increases the

[1] https://www.slicer.org/slicerWiki/index.php/Documentation/4.4/SlicerApplication/ExtensionsManager

performance of the tissue response calculation by utilizing two formulations on a single mesh: a linear formulation for fast computation, albeit less accurate, and a non-linear formulation for more realistic simulations near the interaction area. It then couples these two formulations seamlessly by updating the feedback forces at the interface of the two formulations [26]. With AVM surgery simulation involving various tissues with different physical properties, the hybrid FEM promises improvements in performance on the physics engine by up to 50%. Figure 7 shows preliminary results of our interactive simulator.


Courtesy of TAAFonline.org
Courtesy of clinical-mri.com
AVM Pathology
AVM-Nidus Model
Mechanical: 30K Tetrahedrons
Collision: 10K Triangles
Visual: 10K Triangles
Bipolar Cautery Tool
CAD Model: 1832 triangles
Collision Detection - Force Feedback

(a) AVM pathology (b) AVM model (c) Cautery tool (d) Collision detection-interaction

Fig. 7. Interactive simulator for neurosurgical procedures involving brain vasculature, in SOFA. The simulator uses a CBC3D plugin to generate a volumetric mesh of the vascular structures.

5. Results

We evaluated the CBC3D method on five segmented images. Table 2 lists the image data of this study. Cases 1-3 are anisotropic (non-uniform spacing); cases 4-5 are isotropic (uniform spacing). Case 1 is a single material geometry, while cases 2-5 are multi-material geometries. We compared our implementation with two other open-source codes: CGAL v4.5.2 and Cleaver v1.5.4.

The CGAL is a meshing engine based on a Delaunay refinement scheme [27]. It first constructs a boundary mesh from the image and then a volume mesh given the surface mesh as an input. Before the 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 refinement process is driven by criteria concerning either the size or shape of mesh cells and surface facets. It terminates when there are no more mesh cells or surface facets violating the criteria. The refinement is followed by a mesh optimization phase [8] to remove slivers and provide a good quality mesh. Table 3 lists the CGAL input parameters used in this study.

Cleaver [11] is a multi-material lattice-based meshing method. Initially, it constructs a high-quality BCC lattice that covers the input image. Next, it locally warps or cuts the lattice so it will conform to multi-material arbitrary surfaces. The violation parameters $\alpha_{short}, \alpha_{long}$ decide the trade-off between snapping/warping and stencil cleaving; hence, they implicitly control the size of the mesh. In this study, we use the default violation parameters (Table 3) that achieve a worse case minimum dihedral angle of 2.76° and worst case maximum dihedral angle of 175.42°.

Table 3 lists CBC3D's input parameters. We chose a lattice spacing approximately equal 8-10 times the input image spacing, and a fidelity value $F = 0.95$ for the refinement criterion (subsection 3.2). Finally, we used an "edge" non-connectivity pattern and five deformation iterations to balance the mesh fidelity and the speed of the deformation.

Table 2. Segmented image data of this study.

Case	Type	Tissues	Image Spacing (mm)	Image Size (voxels)
1	Anisotropic	Brain	$0.85 \times 0.85 \times 2.50$	$286 \times 286 \times 90$
2	Anisotropic	Brain-Nidus	$0.85 \times 0.85 \times 0.72$	$256 \times 256 \times 195$
3	Anisotropic	Brain-Tumor	$0.48 \times 0.48 \times 1.00$	$384 \times 512 \times 186$
4	Isotropic	Brain-Ventricles	$1.00 \times 1.00 \times 1.00$	$240 \times 240 \times 134$
5	Isotropic	Brain-AVM	$0.87 \times 0.87 \times 0.87$	$256 \times 256 \times 134$

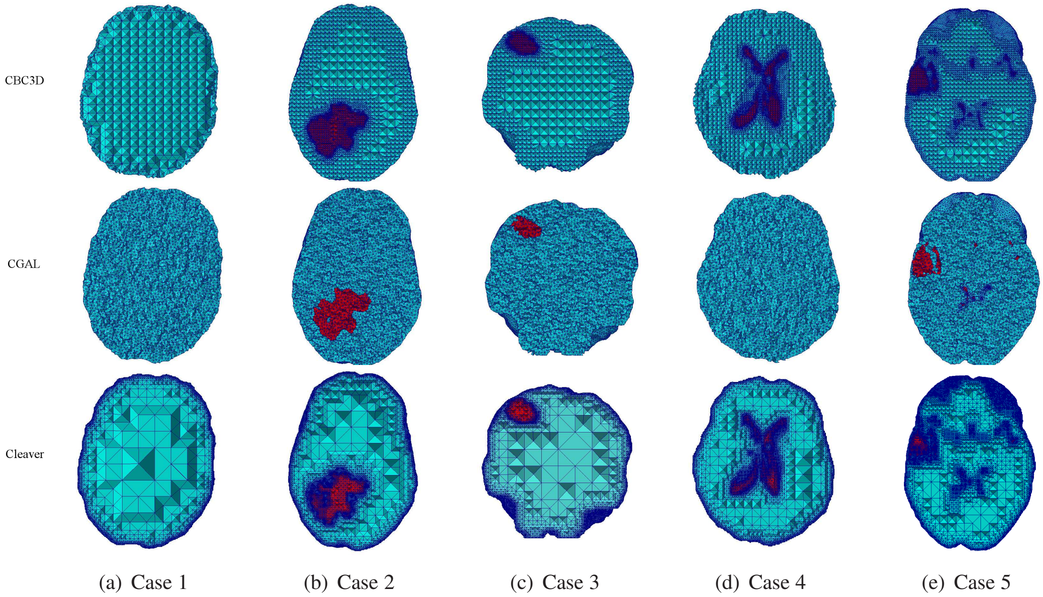
5.1. Qualitative Results

Mesh gradation. Figure 8 depicts cross sections of all the generated tetrahedral meshes in this study. Each cross section illustrates the mesh gradation and the conformity between the sub-meshes of different tissues (cyan and red).

Table 3. Input parameters for the experiments in this study.

Method	Parameter	Value	Description
CBC3D	BCC_{size}	8	lattice spacing (mm)
	F	0.95	fidelity to tissue boundaries
	$nonConnectivity$	"edge"	pattern for the target point extraction
	N_{iter}	5	number of deformation iterations
CGAL	$facet\_angle$	30	lower bound for the angles (degrees) of the surface mesh facets
	$facet\_size$	2	upper-bound for the radii of the surface Delaunay balls
	$facet\_distance$	2	upper bound for the face distance
	$cell\_radius\_edge\_ratio$	1.5	upper bound for the radius-edge ratio of the tetrahedra
	$cell\_size$	2.5	upper-bound for the circumradii of the tetrahedra
Cleaver	α_{short}	0.357	threshold along a diagonal edge for nonviolating edge-cuts
	α_{long}	0.203	threshold along an axis-aligned edge for nonviolating edge-cuts

The CBC3D and Cleaver methods generate graded meshes consisting of smaller elements near the tissue boundaries/interfaces, and larger elements in the homogeneous regions. Cleaver exhibits a more aggressive gradation than CBC3D. On the other hand, CGAL, for the specific input parameters, produces meshes without any variation on the tetrahedral size near the tissue interfaces. CGAL ignores the ventricle material in case 4, and thus, it generates a single-tissue mesh (column 8(d)).



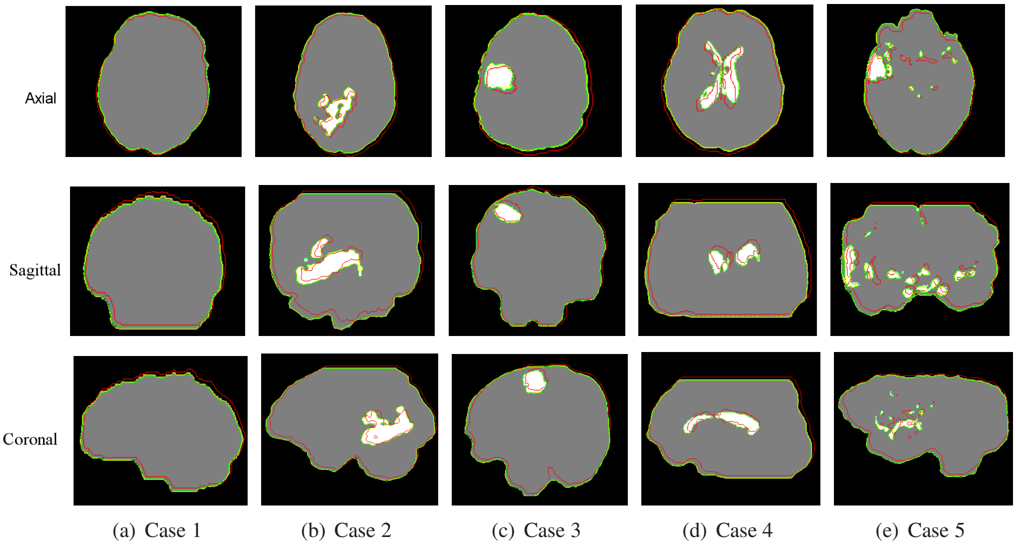
(a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4 (e) Case 5

Fig. 8. Qualitative results on mesh gradation. Each row depicts cross sections of tetrahedral meshes from a single method. Each column represents a single case. In each case we use the same cross section plane for all the methods. Cyan represents the sub-mesh of the brain tissue. Red represents the sub-mesh of the other tissue.

Geometric mesh fidelity. Figure 9 depicts the achieved geometric mesh fidelity on three image slices: axial, sagittal, and coronal. The green, yellow, and red lines, illustrate the intersections between the surface/interface of the CBC3D, CGAL, and Cleaver meshes, respectively, and the image slice. The closer the intersection to the physical boundary/interface, the higher the fidelity. In the specific image slices depicted in Figure 9, the CBC3D (green) and CGAL (yellow) methods exhibit a higher geometric fidelity compared to the Cleaver method (red). The latter seems to produce slightly shifted meshes in relation with the segmented object in the image slices. Because CGAL generates a single-tissue mesh for case 4, the yellow line describing the intersection between the interface of the brain-ventricle mesh and the image slice is missing (column 9(d)).

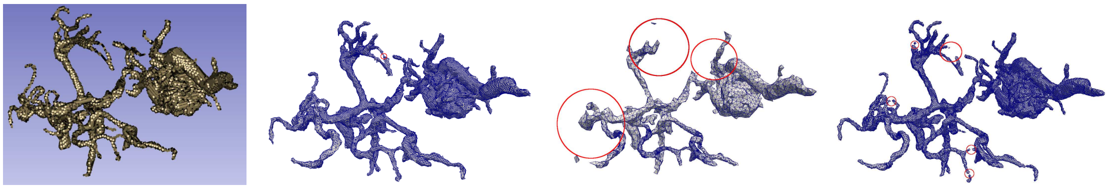
Topologic mesh fidelity. We evaluated the achieved topologic mesh fidelity on a complex AVM pathology (Figure 10(a)). The red circles in Figures 10(b)-10(d) locate the AVM mesh regions with a poor topologic fidelity. The CBC3D demonstrates the highest fidelity among the three methods, preserving almost all the vascular structures in the mesh (Figure 10(b)). CGAL resolves only sufficiently large vessels (Figure 10(c)). Cleaver exhibits a higher topologic fidelity than CGAL, but it does not preserve some thin vessels in five different locations (Figure 10(d)).

Mesh smoothness. Figure 11 compares the smoothness of a brain mesh generated by the three methods. Cleaver captures more features on the brain surface because it generates a larger number of triangles than CBC3D and CGAL. However, Cleaver's mesh exhibits a lower fidelity because it is slightly shifted compared to the position of the brain depicted in the image. The is illustrated clearly in Figure 9(c) and verified by the HD metric in Table 4.



(a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4 (e) Case 5

Fig. 9. Qualitative results on (geometric) mesh fidelity. From top to bottom row: axial, sagittal, and coronal image slices. From left to right column: Case 1-5. Gray represents the brain segmentation. White represents the segmentation of the other tissue. Green, yellow and red illustrate the intersection between the surfaces/interfaces of the CBC3D, CGAL and Cleaver meshes, respectively, and the image slice. The closer the intersection to the physical image boundaries/interfaces, the higher the fidelity on the specific slice.



(a) AVM segmentation (b) CBC3D (c) CGAL (d) Cleaver

Fig. 10. Qualitative results on (topologic) mesh fidelity. (a): AVM extracted from case 5 (Table 2). (b), (c), (d): CBC3D, CGAL, and Cleaver tetrahedral mesh, respectively, for this AVM pathology. The red circles indicate regions where the image topology is not preserved in the mesh.

5.2. *Quantitative Results*

We accessed the achieved mesh quality in terms of the minimum, maximum dihedral angles [28,29]. Ideally, the minimum dihedral angle should be large, and the maximum dihedral angle should be small. We access the achieved mesh fidelity in terms of a publicly available Hausdorff Distance (HD) metric [30]. The smaller the HD value, the higher the fidelity. For a multi-tissue mesh, the HD is computed as: $\mathrm{HD} = \max(\mathrm{HD}_1, \mathrm{HD}_2, \ldots, \mathrm{HD}_n)$, where n is the number of the tissues, HD_i is the two sided Hausdorff Distance of sub-mesh i, and $i = 1, 2, \ldots, n$. HD_i is computed between two point-sets: the first set contains the surface vertices of sub-mesh i; the second set contains the voxels on the surface of tissue i.

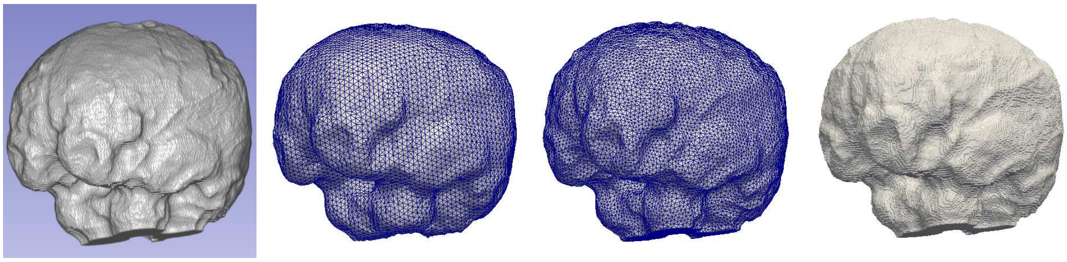

(a) Brain segmentation (b) CBC3D (c) CGAL (d) Cleaver

Fig. 11. Qualitative results on mesh smoothness. (a): Brain segmentation from case 3 (Table 2). (b), (c), (d): CBC3D, CGAL, and Cleaver surface mesh consisting of 18018, 28098, and 443276 triangles, respectively, for this brain segmentation. In (d) we do not visualize the triangle's edges, due to their large number.

Table 4 presents the quantitative results. Cleaver achieves the highest average mesh quality, with (min, max) dihedral angles (12.61°, 148.13°). CGAL's optimization phase aims at eliminating slivers providing good shaped tetrahedra with slightly worse quality than Cleaver. The CBC3D generates meshes of sufficient quality with average (min, max) dihedral angles (4.81°, 172.56°). Our implementation does not provide any theoretical angle bounds for the deformed mesh, but it guarantees a superior element quality during the red-green refinement, with worse (min, max) dihedral angles (30°, 116.5°). In the future, we will introduce an optimization step based on local element transformations and vertex repositioning, to further improve the quality of the smoothed mesh.

Our meshing method achieves the highest fidelity in this study. In terms of a Hausdorff Distance metric, CBC3D exhibits on average 3.79 and 1.32 times higher fidelity compared to CGAL and Cleaver, respectively (Table 4). CGAL achieves a low fidelity in case 4 (46.54 mm), and case 5 (21.35 mm) because, in the former, it ignores the lateral ventricles (Figures 8(d), 9(d)), and in the latter, it does not accurately preserve the vessels structures (Figure 10(c)). Also, CBC3D keeps the element count low. It generates on average about 1.1 times more, and 6.3 times fewer elements compared to CGAL and Cleaver, respectively (Table 4). CBC3D and CGAL provide an explicit control on the element size, via parameters BCC_{size}, and $facet\_size$, $cell\_size$, respectively. On the other hand, Cleaver implicitly controls the mesh size with the violation parameters $\alpha_{short}, \alpha_{long}$ (Table 3).

Table 4. Quantitative evaluation results. The dihedral angle ∈ (0°, 180°) denotes the mesh quality. The larger the minimum angle or the smaller the maximum angle, the higher the mesh quality. The HD metric denotes the mesh fidelity. The smaller the HD, the higher the fidelity.

Case	#Tetrahedra			Dihedral angle (min, max)			HD (mm)		
	CBC3D	CGAL	Cleaver	CBC3D	CGAL	Cleaver	CBC3D	CGAL	Cleaver
1	69101	414470	2550628	(12.20°, 157.67°)	(12.01°, 162.27°)	(10.88°, 152.92°)	3.47	3.21	6.38
2	500777	376077	3238752	(0.21°, 179.68°)	(12.06°, 161.86°)	(9.49°, 153.51°)	6.22	7.66	5.42
3	294005	516927	3276161	(0.29°, 179.57°)	(12.03°, 163.02°)	(11.34°, 153.15°)	4.11	3.37	5.03
4	541943	346337	2629208	(11.18°, 166.19°)	(5.43°, 170.53°)	(25.06°, 126.94°)	3.02	46.54	5.56
5	952986	350449	3235942	(0.18°, 179.72°)	(5.42°, 171.07°)	(6.23°, 154.14°)	4.87	21.35	6.32
Average	471762	400852	2986138	(4.81°, 172.56°)	(9.39°, 165.83°)	(12.61°, 148.13°)	4.33	16.42	5.74

In this study, all the experiments were conducted on a machine with an Intel i7-2600@3.40 GHz CPU, and 16 GB of RAM. For a relatively small multi-tissue mesh (case 3, Table 4), the CBC3D completes in 272.88 seconds (i.e. 63.59 + 208.70 + 0.59 seconds for the mesh generation & refinement, deformation, and I/O, respectively). For a larger multi-tissue mesh (case 5, Table 4), the CBC3D completes in 530.39 seconds (i.e. 94.56 + 434.10 + 1.73 seconds for the mesh generation & refinement, deformation, and I/O, respectively). This CBC3D version does not address any kind of performance issues in terms of time efficiency. In our future work, we will improve the CBC3D execution time by introducing a Structured Adaptive Mesh Refinement (SAMR) scheme [31], and we will conduct an extensive performance evaluation on the three meshing methods.

In summary, from our previous work on lattice-based meshing, we showed that quality, fidelity and size criteria, are in conflict (see Table 1 in [12]). This qualitative/quantitative evaluation shows that among the methods in this study, the CBC3D meshes: (1) represent most accurately the geometry/topology of the image object, (2) keep the element count low, and (3) exhibit an acceptable element quality; thus, are suitable for anatomic modeling of AVM for interactive surgical simulations.

6. Summary and Conclusion

We presented an open-source implementation of an adaptive multi-tissue tetrahedral mesh generator customized for anatomic modeling of AVM for surgical simulations. Our approach, initially, generates an adaptive BCC mesh of high quality elements, from a multi-labeled segmented image. Then, it deforms the mesh surfaces to their corresponding physical image boundaries, to improve the mesh fidelity and smoothness. We introduced a new implementation of a mesh deformation scheme, which builds upon the ITK toolkit, improving the overall reliability and portability of our method. The deformation scheme relies on a multi-material, point-based registration, and uses non-connectivity patterns to implicitly control the number of the extracted image features needed for the registration. As a result, it balances the trade-off between the achieved mesh fidelity and the deformation speed. Our software is available as a stand-alone ITK library (multi-tissue version), as an extension in 3D Slicer (single-tissue; the multi-tissue code will become available before the IMR meeting), and as a SOFA plugin within an interactive simulator for neurosurgical procedures involving brain AVM (multi-tissue).

We compared our implementation with two other open-source meshing codes: CGAL v4.5.2 and Cleaver v1.5.4. We reported qualitative/quantitative results on the geometric/topologic fidelity, the mesh gradation, smoothness and quality, obtained from five isotropic/anisotropic images. The evaluation indicated that our technology provides well graded, smooth meshes, of reasonable size, that reflect a certain degree of visual reality, without compromising the geometric/topologic fidelity of the anatomic modeling. The CBC3D meshes accurately conform to the object's surfaces/interfaces, and preserve the majority of the image features in complex geometries pertinent to AVM. In terms of a Hausdorff Distance metric, the CBC3D exhibits on average 3.79 and 1.32 times higher fidelity, compared to CGAL and Cleaver methods, respectively. Qualitative results are consistent with these data.

Additionally, we will introduce a Structured Adaptive Mesh Refinement (SAMR) scheme [31] to improve the performance and the gradation of the CBC3D method. SAMR, initially discretizes the image domain with a uniform mesh, and then it generates new finer sub-meshes (components) near the areas where the physics are "changing" (e.g. AVM blood flow simulation in regions of interest). SAMR uses an independent PDE solver in each of the mesh components, and accesses 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. Besides, 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, where each mesh component is associated to a single core/node and the under-utilized cores/nodes (e.g. those that compute the PDE solution in smaller number of cells or mesh points) can request automatically additional work from the rest of the cores/nodes.

Acknowledgements

Research reported in this publication was supported in part by the Modeling and Simulation Fellowship at Old Dominion University, the Office of The Director, National Institutes Of Health under Award Number R44OD018334 the NSF grants: CCF-1139864 and CCF-1439079 and by the Richard T.Cheng Endowment. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH and NSF. Special thanks for helping with edits and proofreading to Alexis Brueggeman.

References

[1] I.-F. Talos, N. Archip, Volumetric non-rigid registration for mri-guided brain tumor surgery (2007).

[2] N. Archip, O. Clatz, S. Whalen, D. Kacher, A. Fedorov, A. Kot, N. Chrisochoides, F. Jolesz, A. Golby, P. M. Black, S. K. Warfield, 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 (2007) 609–624.

[3] Y. Liu, A. Kot, F. Drakopoulos, C. Yao, A. Fedorov, A. Enquobahrie, O. Clatz, N. P. Chrisochoides, An itk implementation of a physics-based non-rigid registration method for brain deformation in image-guided neurosurgery, Frontiers in Neuroinformatics 8 (2014).

[4] R. J. Dholakia, F. Drakopoulos, C. Sadasivan, X. Jiao, D. J. Fiorella, H. H. Woo, B. B. Lieber, N. Chrisochoides, High fidelity image-to-mesh conversion for brain aneurysm/stent geometries, in: IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2015.

[5] D. Boltcheva, M. Yvinec, J.-D. Boissonnat, Mesh generation from 3d multi-material images, in: G.-Z. Yang, D. Hawkes, D. Rueckert, A. Noble, C. Taylor (Eds.), Medical Image Computing and Computer-Assisted Intervention MICCAI 2009, volume 5762 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2009, pp. 283–290.

[6] J.-P. Pons, F. Sgonne, J.-D. Boissonnat, L. Rineau, M. Yvinec, R. Keriven, High-quality consistent meshing of multi-label datasets, in: N. Karssemeijer, B. Lelieveldt (Eds.), Information Processing in Medical Imaging, volume 4584 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2007, pp. 198–210.

[7] P. A. Foteinos, N. P. Chrisochoides, High quality real-time image-to-mesh conversion for finite element simulations, Journal of Parallel and Distributed Computing 74 (2014) 2123–2140.

[8] S.-W. Cheng, T. K. Dey, H. Edelsbrunner, M. A. Facello, S.-H. Teng, Silver exudation, J. ACM 47 (2000) 883–904.

[9] L. P. Chew, Guaranteed-quality delaunay meshing in 3d (short version), in: Proceedings of the Thirteenth Annual Symposium on Computational Geometry, SCG '97, ACM, New York, NY, USA, 1997, pp. 391–393.

[10] F. Labelle, J. R. Shewchuk, Isosurface stuffing: Fast tetrahedral meshes with good dihedral angles, ACM Trans. Graph. 26 (2007).

[11] J. Bronson, J. Levine, R. Whitaker, Lattice cleaving: A multimaterial tetrahedral meshing algorithm with guarantees, Visualization and Computer Graphics, IEEE Transactions on 20 (2014) 223–237.

[12] A. N. Chernikov, N. P. Chrisochoides, Multitissue tetrahedral image-to-mesh conversion with guaranteed quality and fidelity, SIAM Journal on Scientific Computing 33 (2011) 3491–3508.

[13] Y. Zhang, C. Bajaj, B.-S. Sohn, 3d finite element meshing from imaging data, Computer Methods in Applied Mechanics and Engineering 194 (2005) 5083 – 5106. Unstructured Mesh Generation.

[14] Y. Zhang, J. Qian, Resolving topology ambiguity for multiple-material domains, Computer Methods in Applied Mechanics and Engineering 247248 (2012) 166 – 178.

[15] N. Molino, R. Bridson, J. Teran, R. Fedkiw, A crystalline, red green strategy for meshing highly deformable objects with tetrahedra, in: In 12th Int. Meshing Roundtable, 2003, pp. 103–114.

[16] M. Fuchs, M. Wagner, J. Kastner, Boundary element method volume conductor models for eeg source reconstruction, Clinical Neurophysiology 112 (2001) 1400–1407.

[17] R. Radovitzky, M. Ortiz, Tetrahedral mesh generation based on node insertion in crystal lattice arrangements and advancing-front-delaunay triangulation, Computer Methods in Applied Mechanics and Engineering 187 (2000) 543–569.

[18] A. Fedorov, N. Chrisochoides, R. Kikinis, S. Warfield, Tetrahedral mesh generation for medical imaging, in: Int Conf Med Image Comput Comput Assist Interv. MICCAI 2005, volume 8, 2005.

[19] Y. Liu, P. Foteinos, A. Chernikov, N. Chrisochoides, Mesh deformation-based multi-tissue mesh generation for brain images, Engineering with Computers 28 (2012) 305–318.

[20] M. Jenkinson, M. Pechaud, S. Smith, BET2: MR-based estimation of brain, skull and scalp surfaces, in: Eleventh Annual Meeting of the Organization for Human Brain Mapping, 2005.

[21] L. Antiga, M. Piccinelli, L. Botti, B. Ene-Iordache, A. Remuzzi, D. Steinman, An image-based modeling framework for patient-specific computational hemodynamics, Medical & Biological Engineering & Computing 46 (2008) 1097–1112.

[22] W. E. Lorensen, H. E. Cline, Marching cubes: A high resolution 3d surface construction algorithm, in: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '87, ACM, New York, NY, USA, 1987, pp. 163–169.

[23] S. Curtis, R. Tamstorf, D. Manocha, Fast collision detection for deformable models using representative-triangles, in: Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games, I3D '08, ACM, New York, NY, USA, 2008, pp. 61–69.

[24] M. Tang, S.-E. Yoon, D. Manocha, Adjacency-based culling for continuous collision detection, Vis. Comput. 24 (2008) 545–553.

[25] M. Tang, D. Manocha, R. Tong, Fast continuous collision detection using deforming non-penetration filters, in: I3D '10: Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games, ACM, New York, NY, USA, 2010, pp. 7–13.

[26] V. Arikatla, R. Ortiz, D. Thompson, D. Adams, A. Enquobahrie, S. De, A hybrid approach to simulate tissue behavior during surgical simulation, in: 4th International Conference on Computational and Mathematical Biomedical Engineering- CMBE2015, 2015.

[27] J. R. Shewchuk, Tetrahedral mesh generation by delaunay refinement, in: Proceedings of the Fourteenth Annual Symposium on Computational Geometry, SCG '98, ACM, New York, NY, USA, 1998, pp. 86–95.

[28] J. R. Shewchuk, What is a good linear element? - interpolation, conditioning, and quality measures, in: In 11th International Meshing Roundtable, 2002, pp. 115–126.

[29] O. Goksel, S. Salcudean, High-quality model generation for finite element simulation of tissue deformation, in: G.-Z. Yang, D. Hawkes, D. Rueckert, A. Noble, C. Taylor (Eds.), Medical Image Computing and Computer-Assisted Intervention MICCAI 2009, volume 5762 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2009, pp. 248–256.

[30] F. Commandeur, J. Velut, O. Acosta, A vtk algorithm for the computation of the hausdorff distance, The VTK Journal (2011).

[31] S. Baden, N. Chrisochoides, D. Gannon, M. Norman, Structured Adaptive Mesh Refinement (SAMR) Grid Methods, Springer-Verlag New York, 2000.