2017

Homeomorphic Tetrahedral Tessellation for Biomedical Images

Jing Xu
Old Dominion University

Andrey N. Chernikov
Old Dominion University

Original Publication Citation

Xu, J., & Chernikov, A. N. (2017). *Homeomorphic tetrahedral tessellation for biomedical images*. Paper presented at the Modeling, Simulation, and Visualization Student Capstone Conference, Suffolk, VA, April 20, 2017. https://sites.wp.odu.edu/capstone2018/wp-content/uploads/sites/6330/2017/11/CAPSTONE2017_Proceedings.pdf

HOMEOMORPHIC TETRAHEDRAL TESSELLATION FOR BIOMEDICAL IMAGES

Jing Xu

Department of Computer Science
Old Dominion University
4700 Elkhorn Ave
Norfolk, VA, USA
jxu@cs.odu.edu

Andrey N. Chernikov

Department of Computer Science
Old Dominion University
4700 Elkhorn Ave
Norfolk, VA, USA
achernik@cs.odu.edu

ABSTRACT

We present a novel algorithm for generating three-dimensional unstructured tetrahedral meshes for biomedical images. The method uses an octree as the background grid from which to build the final graded conforming meshes. The algorithm is fast and robust. It produces meshes with high quality since it provides dihedral angle lower bound for the output tetrahedra. Moreover, the mesh boundary is a geometrically and topologically accurate approximation of the object surface in the sense that it allows for guaranteed bounds on the two-sided Hausdorff distance and the homeomorphism between the boundaries of the mesh and the boundaries of the materials. The theory and effectiveness of our method are illustrated with the experimental evaluation on synthetic and real medical data.

Keywords: tetrahedralization, quality, fidelity, homeomorphism

1 INTRODUCTION

With medical data sets, one can generate conforming quality meshes of the spatially realistic domains that help producing computer aided visualization, manipulation, and quantitative analysis of the multi-dimensional image data. The domain of focus often possesses heterogeneous materials that specify functionally different characteristic properties, so it is usually segmented into multiple regions of interest (materials). In finite element analysis, each material of interest is assigned an individual attribute or a material coefficient. Thus, meshes with conforming boundaries describing each of the partitioned material regions are generated for these purposes.

The generation of geometric discretizations from segmented multi-material images presents many challenges. In particular, a mesh should meet constraints on both the shape and the size of its elements, and must conform at the material boundaries. In addition, the algorithm must handle arbitrary topology. In this paper we present an algorithm for constructing tetrahedral volume meshes that are suitable for real-time finite element analysis, i.e., they satisfy the following criteria:

1. Elements with arbitrarily small angles which cause the stiffness matrix in FE analysis to be ill-conditioned do not appear in the mesh. Specificly, we guarantee that all dihedral angles are above a user-specified lower bound which can be set to any value up to 19.47°.

Capstone-<Symposium Acronym>, 2017 April 20, Suffolk, VA, 23435, USA;

2. The mesh offers a reasonably close representation (fidelity) of the underlying materials. In particular, the two-sided Hausdorff distance between the boundaries of the mesh and the boundaries of the materials respects the user specified fidelity bounds.

3. The mesh is able to offer a faithful topology of the materials. In other words, mesh boundary is homeomorphic to the object surface. We discuss the concept of homeomorphism, and give a sufficient condition for the approximation to offer homeomorphism.

4. The mesh contains a small number of elements that comply with the three guarantees above.

There has been a significant amount of work on guaranteed quality mesh construction. One approach for generating tetrahedral meshes of smooth and piecewise smooth surfaces is that the input is assumed to be an implicit function $f : R^3 \rightarrow Z$ such that points in different regions of interest evaluate f differently. One guaranteed-quality technique is based on the Delaunay refinement (Boissonnat and Oudot 2005, Cheng et al. 2010, Foteinos et al. 2014). In Foteinos et al. (2014)'s work, the authors present a Delaunay meshing algorithm with several mathematical guarantees. They proved that the tetrahedra in the output mesh have the radius-edge ratio less than 1.93. The two-sided Hausdorff distance between the object surface and mesh boundary is bounded by a user-specified parameter. Using a strategy called ε-*Sample* (Amenta et al. 2000, Amenta et al. 2003), the mesh boundary is proved to be ambient isotopic to the object surface. However, the method only supplies the circumradius-to-shortest-edge ratio bound. Even if this ratio is very small, it can not avoid the almost flat tetrahedra. Another guaranteed-quality technique employs a space-tiling background grid to guide the creation of a mesh (Labelle and Shewchuk 2007, Bronson et al. 2014, Chernikov and Chrisochoides 2011), the focus of this paper. Isosurface Stuffing (Labelle and Shewchuk 2007) is a guaranteed-quality tetrahedral meshing algorithm for general surfaces under the assumption that the surface is a smooth 2-manifold. It offers the one-sided Hausdorff distance guarantee from the mesh to the model. If the surface is a smooth manifold with bounded curvature, it also provides the one-sided Hausdorff distance guarantee from the model to the mesh. Using interval arithmetic, the dihedral angles for the mesh with uniform sized boundary are proved to be bounded between 10.7° and 164.8°. However, our method depends on a standard octree and decimation, and is able to achieve the minimum dihedral angle bound 19.47°.

This work builds upon the Lattice Decimation (LD) method (Chernikov and Chrisochoides 2011). LD is a tetrahedral image-to-mesh conversion algorithm that allows for guaranteed bounds on the smallest dihedral angle (up to 35.26°) and on the Hausdorff distance between the boundaries of the mesh and the boundaries of the materials. The algorithm constructs an octree and splits the octree until no leaf contains voxels from multiple materials. Then it fills the octree leaves with high-quality elements. This initial mesh has a large number of elements because it satisfies the highest dihedral angle and fidelity bounds. The authors designed a post-processing decimation step using vertex removal operation. It coarsens the mesh to a much lower number of elements while at all times maintaining the required fidelity and quality bounds. However, the decimation step is a greedy algorithm which was not designed for smooth transition in element size. In fact, it can produce clusters of small elements surrounded by much larger elements. In this work, we refine the octree to a lower level and the element size does not have so big difference, therefore, the issue can be mitigated. However, the relatively coarse mesh brings a new issue: the topological faithfulness may not be guaranteed. We designed a new technique called single manifold condition that solves this problem.

In this paper, we approximate the boundary of the materials with a set of triangular patches in octree leaves using a pre-defined surface look-up table. The triangular patches all together form a waterproof surface mesh which is homeomorphic to the boundary of the materials by proof. We achieve the two-sided Hausdorff distance bound by constructing a sequence of such surface meshes until the fidelity condition in each leaf is satisfied. The octree leaves are filled with high-quality elements from a pre-defined volume look-up table.

During decimation, the initial mesh is coarsened to a much lower number of elements while at all times the quality, fidelity and topological requirements are maintained.

2 METHODOLOGY

The proposed algorithm is described as follows: the algorithm takes a two- or a three-dimensional multi-material image as its input. The user also specifies as input the target fidelity bounds (two-sided Hausdorff distance) and the desired angle lower bound. The algorithm outputs a quality mesh which is a good geometric and topological approximation of the underlying object. We first define some concepts relative to the topological faithfulness.

Definition 2.1. Homeomorphism A topological notion of equivalence. A mapping $\mu : X \to Y$ defines a homeomorphism between two compact Euclididean subspace X and Y if μ is continuous, one-to-one and onto. The inverse function μ^{-1} is also continuous.

Definition 2.2. Single manifold condition The intersection of the image boundary with each of the boundary leaves is a $(n-1)$-manifold with boundary, n being the dimension.

Specificly, in two dimensional case, the image boundary edges form a 1-manifold with boundary, a chain composed by image boundary edges. On this chain, every image boundary vertex has two neighbors, except the first and last ones, which have only one neighbor. In three dimensional case, the degree of the image boundary edge is defined by the number of boundary faces that incident upon the edge. The image boundary faces form a 2-manifold with boundary, a sheet on which the degree one image boundary edges form a cycle and all the other image boundary edges are degree two edges.

2.1 Initial mesh construction

The algorithm first constructs an octree that completely encloses all the materials (except for the background voxels) from the bitmap. The boundaries between the octree leaves correspond exactly to the boundaries between the voxels. Besides that, an extra space between the materials and the exterior boundaries of the octree should be equal to or larger than the user specified fidelity bounds. Initially, the algorithm splits the octree such that the leaves satisfy the single manifold condition, and the sizes of the leaves respect the 2-to-1 ratio. Then it generates a waterproof surface mesh which is homeomorphic to the the boundaries of the materials from a pre-defined surface look-up table. It computes the two-sided Hausdorff distance between the boundaries of the mesh and the boundaries of the materials in each leaf. It splits the leaf into 8 children if the Hausdorff distance of either side is larger than the user specified fidelity bounds. If at least one of the leaves was split because of the conflicting of the fidelity condition, the surface mesh is discarded. The algorithm iteratively checks the single manifold condition and the 2-to-1 ratio, and generates waterproof surface meshes until the two-sided Hausdorff distance respects the user specified fidelity bounds.

2.1.1 Waterproof surface mesh

When the octree leaves respect the single manifold condition and the 2-to-1 ratio, the algorithm generates triangular patches from each octree leaf such that all the patches form a waterproof surface mesh. To generate the waterproof surface mesh, we use an approach reminiscent of the Marching Cubes algorithm (Lorensen and Cline 1987). In contrast to the Marching Cubes algorithm, our algorithm evaluates the vertices to three values: positive, negative and zero. A vertex of a leaf is evaluated to be positive if the vertex is located inside a material, to be negative if the vertex is located outside the material, and to be zero if the vertex is located exactly on the boundary of the materials. The templates on cubes would be cumbersome for our algorithm, because there would be 3^8 cases need to be analyzed. Instead, we designed a surface look-up

table on squares for each cube face, so there are totally only 3^4 entries in the table. The surface look-up table generates intersection edges on the cube faces, and the triangular patches are generated by connecting those intersection edges with the center of the cube (see Figure 2a). We list all possible stencils in Figure 1 by grouping cases with the opposite relations to the vertex value in all corners into one case and also grouping rotationally symmetric cases.

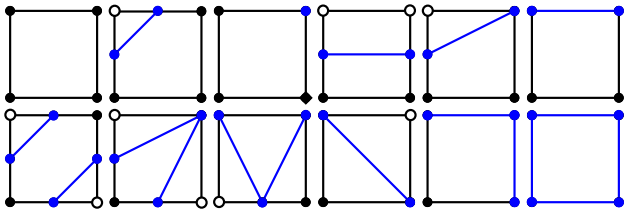
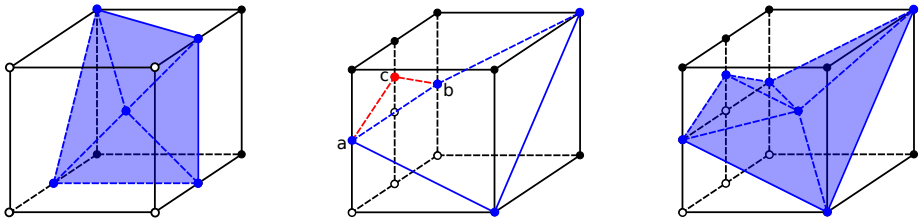


Figure 1: All possible stencils for creating intersection vertices and intersection edges by grouping cases with the opposite relations to the vertex value in all corners into one case and also grouping rotationally symmetric cases. Black circles show the positive vertices, white circles show the negative vertices, and blue circles show the zero vertices. Blue segments show the intersection edges created by the algorithm.



(a) Fan-like triangular patches

(b) Inconsistency between neighboring cubes

(c) Consistency is considered between neighboring cubes

Figure 2: An illustration of generating triangular patches to approximate the image boundary in a leaf. Blue circles show the zero vertices, blue edges on cube faces show the intersection edges and blue triangles show the triangular patches.

One important issue that needs extra concern is the consistency between neighboring leaves. In the case that a face of the octree leaf shared by four smaller neighboring cubes, the consistency is not always guaranteed. For example, in Figure 2b we show a leaf whose left face is shared by four smaller neighboring cubes. From the leaf side of the view, the intersection edge generated for the left face of the leaf should be edge $\overline{ab}$ (shown in blue dashed segment). However, from the four small neighbors side of the view, the intersection edges generated for the same face should be edge $\overline{ac}$ and edge $\overline{cb}$ (shown in red dashed segments). To solve this problem, we process the octree leaves in the order of their size, starting with the smallest. We duplicate the intersection edges from the processed neighbors (including the neighbors of same size and of smaller size), and generate new intersection edges for the faces that shared by the neighboring cubes which have not been processed (see Figure 2c). Then we check if the intersection edges from the six cube faces form a cycle. If all the intersection edges do not form a cycle, the leaf needs to be split. By this way, the surface mesh of our algorithm is guaranteed to be water-tight.

2.1.2 Two-sided Hausdorff distance

The mesh has to provide a close approximation of the object shape. We measure the closeness by the fidelity tolerance, quantified by the two-sided Hausdorff distance from the mesh to the image and the image to the mesh. For image boundary I and mesh boundary M, the one-sided distance from I to M is given by

$$h(I,M) = \max_{i \in I} \min_{m \in M} d(i,m),$$

where $d(\cdot,\cdot)$ is the regular Euclidean distance. The one-sided distance from M to I is given similarly by

$$h(M,I) = \max_{m \in M} \min_{i \in I} d(m,i).$$

The two-sided distance is:

$$H(I,M) = \max\{h(I,M), h(M,I)\}.$$

To measure the Hausdorff distance from the surface mesh to the boundaries of the image, we compute the Euclidean distance transform (Maurer, Qi, and Raghavan 2003) (EDT) of the extended image (same size as the octree), and split the octree until no leaf has the distance of EDT both larger and within the input fidelity bound. We mark the leaves that are within the input fidelity tolerance. If one of the triangular facets of the surface mesh in the leaf intersects at least one of the leaves that marked as outside the fidelity tolerance, the fidelity condition is violated and the leaf is split. To measure the Hausdorff distance from the boundaries of the image to the corresponding surface mesh, we compute the shortest distance from each image boundary vertex in the leaf to the triangular patches of the surface mesh in the leaf. If one of the image boundary vertices has a shortest distance larger than the fidelity tolerance, the fidelity condition is violated and the leaf is split.

2.1.3 Filling in the octree

When the waterproof surface mesh respects the user specified fidelity bounds, the octree is constructed. We fill the octree leaves with high quality tetrahedra using the pre-defined volume look-up table. The volume look-up table is designed based on the same idea as the surface look-up table, however, instead of being used to generate edges on cube faces, it is used to generate triangles on cube faces. The template triangles of the volume look-up table should respect the intersection edges of the surface look-up table, which means that the intersection edges should be the edges of the template triangles. By analyzing all possible shapes of the initial tetrahedra filling a cubic leaf of the octree, the minimum dihedral angle bound is 19.47°.

2.2 Mesh decimation

Similar to the LD method (Chernikov and Chrisochoides 2011), the vertex removal operation is used to coarsen the mesh. We maintain a queue of mesh vertices that are candidates for merging. A vertex is merged to a destination vertex if the vertex and the edge between the vertex and its destination are removed from the mesh. As a consequence, all tetrahedra (triangles) incident upon the removed edge are also removed from the mesh. The detailed operation can be consulted in the paper (Chernikov and Chrisochoides 2011), here we only discuss the merging conditions. A vertex can not be merged along an edge to another vertex if it violates the following requirements:

1. The quality requirement, i.e., if at least one of the newly created elements, as a result of a sequence of merges, is inverted or its dihedral angle is smaller than the input quality angle bound, the merge is discarded.

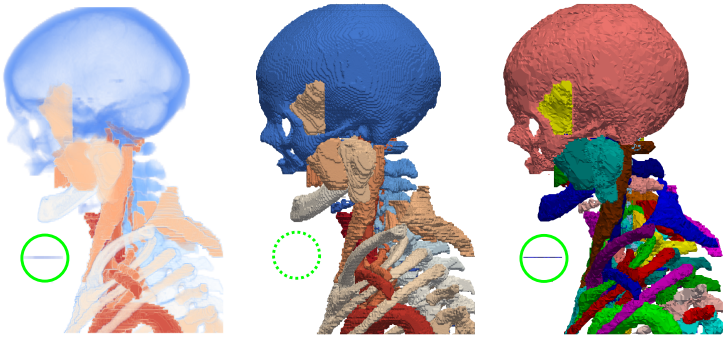
2. The fidelity requirement, i.e., if at least one of the newly created mesh boundary facets has at least one-sided Hausdorff distance larger than the input fidelity bound, the merge is discarded. Same as the fidelity check of the octree construction, this check also consists of two parts, for each of the one-sided Hausdorff distances.

3. The topological equivalence requirement, i.e., if the homeomorphism is not maintained during an operation of merge, the merge is discarded. We apply the following rules to maintain the original structure of the inter-material boundaries: (1) boundary vertices only merge to boundary vertices, (2) a vertex cannot merge to a non-boundary vertex of a different material, (3) for each boundary vertex we also maintain a cumulative list of the octree leaves it belongs to. The merge happens if the union of the set of octree leaves of the merged vertex and the set of the octree leaves of the destination respect the single manifold condition, i.e., the image boundary edges (or faces) of the union form only one $(n-1)$-manifold with boundary, and (4) each tetrahedron keeps the original color even after it changes shape due to vertex merge.

3 EXPERIMENTAL RESULTS

We apply the proposed octree refinement and decimation algorithm (ORD) to both synthetic and real medical data in the following sections. All the experiments were conducted on a 64 bit machine equipped with two 3.06 GHz 6-Core Intel Xeon CPU and 64 GB main memory. The algorithm was implemented in C++, in both two and three dimensions.

For the 3D visualization of the final meshes, we used ParaView (Ahrens et al. 2005), an open source visualization software. In Figure 3, we compare the final mesh generated by a Delaunay open source mesh generator Computational Geometry Algorithms Library (CGAL) and our ORD method on topology on *head neck* image. The size of the *head neck* is $255 \times 255 \times 229$ voxels. Each voxel has side lengths of 0.97, 0.97, and 1.4 units in x, y, and z directions. It is clear that there are two manifolds in the original image (see Figure 3a), however, CGAL only generated one manifold (see Figure 3b). We highlight the missing manifold of CGAL mesh in green in Figure 3a. On the contrary, our method generated two manifolds (see Figure 3c).

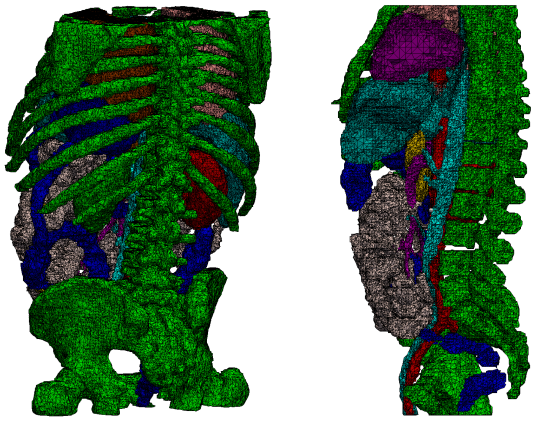


(a) *head neck* image (b) Final mesh of CGAL (c) Final mesh of ORD

Figure 3: The topology comparison of the ORD mesh and the CGAL mesh on *head neck*.

Figure 4 shows the final mesh produced on *abdomen* atlas and its corresponding cut view. The size of the *abdomen* atlas is $512 \times 512 \times 219$ voxels, each voxel has side lengths of 0.96, 0.96, and 2.4 units in x, y, and z directions. In Table 1 we show the comparison of the output mesh size of the ORD and LD for the *abdomen* atlas. We fixed the two-sided Hausdorff distance bound parameters, and vary the dihedral angle bound to the interested value 5°, 10°, 15°, and 19.47° spread through the range of its feasible values (0° to 19.47°).

As we can see from Table 1, the output mesh size is low when $H(I,M)$ is high for all configurations. Also, the final number of tetrahedra decreases as the dihedral angle bound decreases. The tetrahedra generated by ORD before decimation is much fewer than the tetrahedra generated by LD before decimation. When $H(I,M) = 1$, the sizes of ORD meshes are slightly smaller than the sizes of LD meshes. However, when $H(I,M) = 2$, $H(I,M) = 3$ and $H(I,M) = 4$ the ORD algorithm generated significant smaller number of tetrahedra compared to the number of tetrahedra generated by the LD algorithm. We conclude that the ORD algorithm performs better in terms of the number of tetrahedra than LD algorithm even though it maintains the topology.



(a) Final mesh of *abdomen*

(b) A cut view of *abdomen* mesh

Figure 4: The ORD mesh and a cut view of the *abdomen* for $h(I,M) = 2$, $h(M,I) = 2$ and $\theta = 19.47°$.

Table 1: The comparison of final number of tetrahedra for ORD and LD

Hausdorff distance	$h(I,M) = 1, h(M,I) = 1$		$h(I,M) = 2, h(M,I) = 2$		$h(I,M) = 3, h(M,I) = 3$		$h(I,M) = 4, h(M,I) = 4$	
Algorithm	ORD	LD	ORD	LD	ORD	LD	ORD	LD
Before decimation	30,361,224	42,344,304	18,492,773	51,284,042	19,637,745	57,723,164	19,968,161	61,672,648
After decimation with $\theta = 19.47°$	13,027,911	17,608,762	4,062,233	12,434,028	4,164,029	13,672,489	4,243,052	13,238,118
After decimation with $\theta = 15.00°$	10,275,628	11,994,099	1,593,965	3,426,170	1,560,986	3,035,179	1,628,551	2,940,484
After decimation with $\theta = 10.00°$	9,319,252	10,646,685	856,455	2,081,046	825,285	1,592,756	885,394	1,383,768
After decimation with $\theta = 5.00°$	8,510,971	9,488,867	580,192	1,644,226	559,969	1,188,149	604,498	1,079,160

We also conducted an experiment using CGAL and compare the performance with the performance of our ORD algorithm. Table 2 presents the experimental evaluation of the I2M conversion functionality *make_mesh_3* offered by CGAL and by ORD algorithm. We vary the value of parameter *facet_distance*, and show $h(M,I)$ and $h(I,M)$. They are measured by resampled voxel unit. We also list the final number of tetrahedra produced by CGAL, the minimum dihedral angle (measured by degree) and the total running time (measured by seconds). In some cases, when the value of parameter *facet_distance* was increased, the value of $h(M,I)$ also was increased, however, there is no obvious relationship between *facet_distance* and $h(M,I)$. Further more, the values of $h(I,M)$ in all the cases are unreasonably large. We conclude that CGAL can only approximate one-sided Hausdorff distance, while the proposed algorithm can always guarantee that the Hausdorff distance bound is two-sided. CGAL improves mesh properties such as the dihedral angles using various combinations of optimization algorithms, however, we could only obtain the best minimum dihedral angles $5°$. On the contrary, our best best minimum dihedral angle bound is $19.47°$.

Table 2: performance of CGAL

abdomen										
Opt.	*no_lloyd*(), *no_odt*(), *perturb*(), *exude*()					*lloyd*(), *no_odt*(), *perturb*(), *exude*()				
facet_dist.	$h(M,I)$	$h(I,M)$	dih. angle	# of tets	Total time	$h(M,I)$	$h(I,M)$	dih. angle	# of tets	Total time
0.2	3	23	1.49	10,349,057	532.06	6	18	1.61	9,979,982	4558.32
0.4	3	23	2.26	2,042,684	95.80	4	23	1.74	2,000,988	762.00
0.6	4	23	2.35	862,668	39.23	4	24	2.66	849,445	299.09
0.8	4	23	3.05	487,277	27.99	4	24	3.20	481,107	163.09
Opt.	*no_lloyd*(), *odt*(), *perturb*(), *exude*()					*lloyd*(), *odt*(), *perturb*(), *exude*()				
facet_dist.	$h(M,I)$	$h(I,M)$	dih. angle	# of tets	Total time	$h(M,I)$	$h(I,M)$	dih. angle	# of tets	Total time
0.2	5	24	1.01	10,178,967	2132.82	5	18	2.01	9,985,320	5079.16
0.4	7	18	0.86	2,031,462	379.13	5	18	2.17	2,002,825	800.18
0.6	6	19	2.04	862,156	152.53	6	19	2.07	851,196	354.28
0.8	8	19	3.05	487,879	85.16	8	23	5.01	481,937	178.30

4 CONCLUSION

We presented a novel approach for automatic construction of two- and three-dimensional unstructured meshes of multi-material images characterized by (i) guaranteed dihedral angle bound for the output tetrahedra, (ii) guaranteed bounds on two-sided Hausdorff distance between the boundaries of the mesh and the boundaries of the materials, (iii) the mesh boundary is proved to be homeomorphic to the object surface, and (iv) a small number of mesh elements. The applications of the algorithm include mechanical modeling for finite element simulation, computational medicine and computational biology such as medical imaging, image registration, surgical simulation, and image-guided intervention.

5 ACKNOWLEDGMENTS

This work was supported (in part) by the Modeling and Simulation Graduate Research Fellowship Program at the Old Dominion University and NSF grant CCF-1439079. We thank the anonymous reviewers for helpful comments.

REFERENCES

Ahrens, J., B. Geveci, and C. Law. 2005, 01. "ParaView: An End-User Tool for Large Data Visualization.".

Amenta, N., S. Choi, T. K. Dey, and N. Leekha. 2000. "A Simple Algorithm for Homeomorphic Surface Reconstruction". In *Proceedings of the Sixteenth Annual Symposium on Computational Geometry*, SCG '00, pp. 213–222.

Amenta, N., T. J. Peters, and A. C. Russell. 2003. "Computational topology: ambient isotopic approximation of 2-manifolds". *Theoretical Computer Science* vol. 305 (1), pp. 3–15.

Boissonnat, J.-D., and S. Y. Oudot. 2005. "Provably good sampling and meshing of surfaces". *Graphical Models* vol. 67, pp. 405–451.

Bronson, J., J. A. Levine, and R. Whitaker. 2014, Feb. "Lattice Cleaving: A Multimaterial Tetrahedral Meshing Algorithm with Guarantees". *IEEE Transactions on Visualization and Computer Graphics* vol. 20 (2), pp. 223–237.

CGAL. "CGAL, Computational Geometry Algorithms Library". http://www.cgal.org.

Cheng, S.-W., T. K. Dey, and E. A. Ramos. 2010. "Delaunay Refinement for Piecewise Smooth Complexes". *Discrete & Computational Geometry* vol. 43 (1), pp. 121–166.

Chernikov, A., and N. Chrisochoides. 2011. "Multitissue tetrahedral image-to-mesh conversion with guaranteed quality and fidelity". *SIAM Journal on Scientific Computing* vol. 33, pp. 3491–3508.

Foteinos, P., A. Chernikov, and N. Chrisochoides. 2014. "Guaranteed quality tetrahedral Delaunay meshing for medical images". *Computational Geometry Theory and Applications* vol. 47, pp. 539–562.

Labelle, F., and J. R. Shewchuk. 2007. "Isosurface Stuffing: Fast Tetrahedral Meshes with Good Dihedral Angles". *ACM Transactions on Graphics* vol. 26 (3), pp. 57.1–57.10. Special issue on Proceedings of SIGGRAPH 2007.

Lorensen, W. E., and H. E. Cline. 1987. "Marching cubes: A high resolution 3D surface construction algorithm". *COMPUTER GRAPHICS* vol. 21 (4), pp. 163–169.

Maurer, Jr., C. R., R. Qi, and V. Raghavan. 2003, February. "A Linear Time Algorithm for Computing Exact Euclidean Distance Transforms of Binary Images in Arbitrary Dimensions". *IEEE Trans. Pattern Anal. Mach. Intell.* vol. 25 (2), pp. 265–270.