Volumetric Meshes for Real–Time Medical Simulations

Matthias Mueller and Matthias Teschner

Computer Graphics Laboratory
ETH Zurich, Switzerland
Email: muellerm@inf.ethz.ch,
URL: http://graphics.ethz.ch/

Abstract. We present an approach to generating volumetric meshes from closed triangulated surface meshes. The approach initially generates a structured hexahedral mesh based on the bounding box of a surface. Each hexahedron is represented by five tetrahedrons. The complexity of the initial tetrahedral mesh is user-defined. In a second step various subdivision methods can be optionally applied to the tetrahedral mesh. These subdivision steps can adjust the number of tetrahedra and the quality of their aspect ratio. Additionally, they can adapt the accuracy of the correspondence of the boundary of the volumetric mesh with the given surface mesh. A third step removes tetrahedra which are outside the object. And a fourth step optimizes point positions of the resulting volumetric mesh with respect to the aspect ratio of tetrahedra.

The approach has been applied to a variety of medical surface meshes. It is intended to use the generated volume meshes in medical applications, such as real-time simulation for stent placement and for hysteroscopy.

1 State-of-the-Art

There exist various algorithms for tetrahedral mesh generation based on 3D Delaunay triangulation [6], advancing fronts [4], [5], and octrees [7]. Most approaches are focussed on the quality of generated tetrahedra with respect to their aspect ratio or other measures [3]. Detailed surveys of 2D and 3D mesh generation can be found in [1] and [2]. In general, the boundary of the volume mesh equals the given surface mesh and the number of generated volume primitives is comparatively large.

2 Contributions

We present an approach to generating volumetric tetrahedral meshes, which are suitable for real-time physically-based medical simulations. Based on a user-defined initial complexity of the volumetric mesh various subdivision schemes can be applied optionally. These schemes allow to control the final complexity of the tetrahedral mesh. Furthermore, the aspect ratio of tetrahedra and the correspondence of the surface mesh with the boundary of the tetrahedral mesh are controlled by theses schemes. The subdivision schemes can be applied dependent on a desired complexity of the mesh or on a desired quality of the boundary of the volumetric mesh. Low-complexity meshes can be generated for real-time applications and very complex meshes can be computed for simulations where high accuracy is required.

3 Methods

Given a water-tight triangle-based surface mesh we generate a tetrahedral mesh for the contained volume as follows:

In a **first step** we start with a structured mesh of hexahedra that fills the bounding box of the initial surface mesh. The number of hexahedra is user-defined. The bounding box of the surface mesh is filled with a structured mesh of tetrahedra by subdividing each hexahedral cell into five tetrahedra in an alternating fashion.

As a **second step**, we subdivide the tetrahedral mesh along the triangle-based surface mesh. Therefore, three optional steps can be applied. First, for every vertex of the surface mesh we generate a vertex in the initial tetrahedral mesh by subdividing the surrounding tetrahedron into four tetrahedra. Second, if a surface triangle intersects with an edge of the tetrahedral mesh, the edge is subdivided. Third, if an edge of the surface mesh intersects with a face of a tetrahedron, then the tetrahedral face is subdivided. If edges or faces are subdivided, new tetrahedrons are generated accordingly.

All described subdivision processes are optional. When all subdivision steps are applied, the boundary of the tetrahedral mesh equals the triangle-based surface mesh. However, the number of tetrahedra is comparatively large and the aspect ratio of newly generated tetrahedra might be less optimal. If subdivision is not performed, the number of resulting tetrahedra is smaller. However, the boundary of the tetrahedral mesh only approximates the initial surface mesh.

In a **third step** all tetrahedra, that are located outside the surface, are removed. Therefore, for each tetrahedron an arbitrary ray from the center of the tetrahedron to the surface of the bounding volume of the initial surface is considered. If the number of intersections of such a ray with the surface mesh is even, the tetrahedron is considered outside and has to be removed.

As a **fourth step**, the tetrahedral mesh is optimized iteratively. Positions of all vertices are optimized with respect to a set of constraints. Since we are interested in small aspect ratios for all tetrahedra, we use lower and upper distance constraints per edge and per oriented height of each tetrahedron. In addition, all points, that are part of the boundary of the tetrahedral mesh, are constrained to lie on the initial surface mesh. If all subdivision processes in step two have been applied, then all boundary points of the tetrahedral mesh are initially located on the surface mesh. Otherwise, they are forced to move into the direction of the surface mesh by this optimization approach.

4 Results

We have generated volumetric meshes from a variety of surface meshes. The resulting tetrahedral meshes are used in medical applications such as hysteroscopy simulation. Therefore, we have applied the presented approach to a surface model of a uterus consisting of about 5000 triangles. In the initial step we have subdivided the bounding box of the uterus into 1000 hexahedral cells, resulting in 5000 tetrahedra.

If all subdivision methods are applied in the second step, the final mesh consists of 11000 tetrahedra. It takes 15s on a PC to perform all subdivision processes. Optimizing point positions in the third step requires 5s per iteration and 50 optimization steps have been applied.

If subdivision is only performed for surface triangles that interfere with edges of the volumetric mesh, mesh adaptation takes 3s and the final result consists of 5700 tetrahedra. One step to optimize the point positions of the volumetric mesh takes 1s and 20 iterations have been performed in this case.

Dependent on the applied subdivision steps we can vary the number of generated tetrahedra, the quality of aspect ratios, and the quality of the correspondence of the boundary of the volume mesh with the surface mesh.

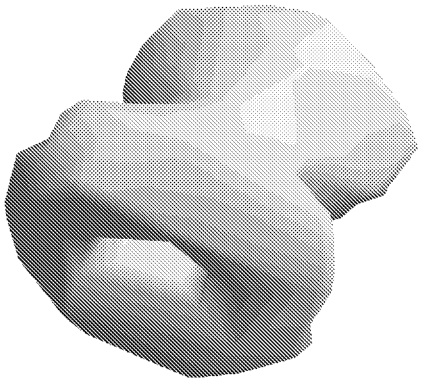


Fig. 1. Surface mesh of a uterus, consisting of 5000 triangle.

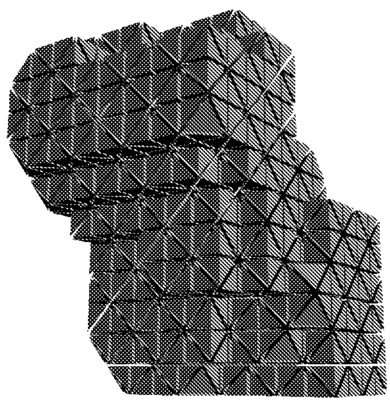


Fig. 2. Structured mesh of hexahedra that approximates the initial surface mesh. The bounding box of the uterus is filled with 5000 tetrahedra by subdividing each hexahedral cell into five tetrahedra. Tetrahedra, that are located outside the surface, are removed. The resulting mesh consists of 3170 tetrahedra.

Fig. 3. In this case, the initial tetrahedral mesh (see Fig. 2) is subdivided along the triangle-based surface mesh. If a surface triangle intersects with an edge of the tetrahedral mesh, the edge is subdivided. New tetrahedrons are generated accordingly. The resulting mesh consists of 7400 tetrahedra.

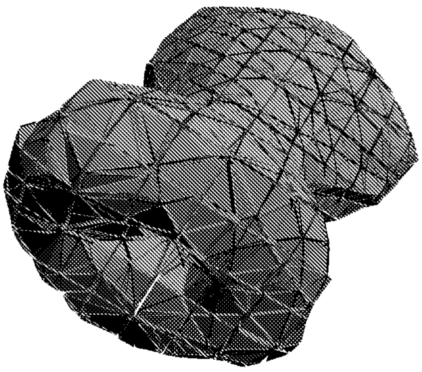
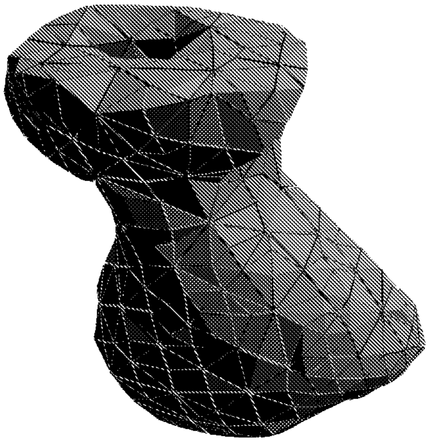


Fig. 4. Optimized mesh. Positions of all vertices (see Fig. 3) are optimized with respect to aspect ratios for all tetrahedra. Further, all boundary points of the tetrahedral mesh are constrained to lie on the initial surface mesh. In addition, tetrahedra with insufficient quality are deleted. The final mesh consists of 2400 tetrahedra.



5 Conclusion

We have presented an approach to generating volumetric tetrahedral meshes from surface meshes, which are suitable for real-time physically-based simulations.

Interior tetrahedra are regularly shaped while tetrahedra near the surface are optimized to meet geometric constraints. Improved aspect ratios of the tetrahedra guarantee a stable numerical solution of the partial differential equations that describe our system. Adaptibility of the number of tetrahedra ensures real-time performance of medical simulations which are applied to the tetrahedral meshes.

Ongoing work focusses on additional postprocessing steps for volumetric meshes where all subdivision steps have been applied. Beside the fact, that the boundary of the volumetric mesh exactly corresponds to the surface mesh, the volumetric mesh can contain small tetrahedra with inappropriate aspect ratios. Therefore, we intend to investigate edge-collaps methods to reduce the number of tetrahedra and to improve the aspect ratio of the remaining tetrahedra, while maintaining the boundary of the volume mesh.

References

1. Marshall Bern and David Eppstein. Mesh Generation and Optimal Triangulation. In *Computing in Euclidean Geometry*, D.-Z. Du and F. Hwang, editors, vol. 1 of Lecture Notes Series on Computing. World Scientific, Singapore, pp.23–90, 1992.
2. Marshall Bern and Paul Plassmann. Mesh Generation. In *Handbook of Computational Geometry*, Joerg Sack and Jorge Urrutia, editors, Elsevier Science, 2000.
3. A. Liu and B. Joe. Relationship between tetrahedron shape measures. *BIT*, vol. 34, pp. 268–287, 1994.
4. R. Loehner and P. Parikh. Three–dimensional grid generation via the advancing–front method. *Int. J. Numer. Meth. Fluids*, vol. 8, pp. 1135–1149, 1988.
5. Nguyen Van Phai. Automatic mesh generation with tetrahedral elements. *Int. J. Numer. Meth. Eng.*, vol. 18, pp. 273–289, 1982.
6. D. F. Watson. Computing the n–dimensional Delaunay–tessellation with application to Voronoi polytopes. *Computer J.*, vol. 24, pp. 167–171, 1981.
7. M. A. Yerry and M. S. Shephard. Automatic three–dimensional mesh generation by the modified–octree technique. *Int. J. Numer. Meth. Eng.*, vol. 20, pp. 1965–1990, 1984.