DelPSC: A Delaunay Mesher for Piecewise Smooth Complexes *

Tamal K. Dey
Department of Computer Science and Engineering
The Ohio State University
Columbus, OH, USA
tamaldey@cse.ohio-state.edu

Joshua A. Levine
Department of Computer Science and Engineering
The Ohio State University
Columbus, OH, USA
levinej@cse.ohio-state.edu

ABSTRACT

This video presents the working of a new algorithm/software called DelPSC that meshes piecewise smooth complexes in three dimensions with Delaunay simplices. Piecewise smooth complexes admit a large class of geometric domains including polyhedra, smooth and piecewise smooth surfaces with or without boundary, and non-manifolds. The algorithm and its proof of correctness are described in the paper [7].

Categories and Subject Descriptors

I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Curve, surface, solid, and object representations*

General Terms

Algorithms, Experimentation, Theory

Keywords

Delaunay refinement, Mesh generation, Piecewise-smooth complexes, Non-smoothness, Non-manifolds

1. INTRODUCTION

Delaunay mesh generation of non-smooth domains such as piecewise smooth surfaces and complexes with provable topological and geometric guarantees is a difficult challenge. The Delaunay refinement technique invented by Chew [6] and Ruppert [9] has been applied to mesh polyhedra [10] and smooth surfaces with provable guarantees [1, 5]. Recently Cheng, Dey, and Ramos [4] proposed a Delaunay refinement algorithm that can mesh a large input class called piecewise smooth complexes (PSCs). This class includes polyhedra, smooth and piecewise smooth surfaces with or without boundary, and even non-manifolds.

*http://www.cse.ohio-state.edu/~tamaldey/delpsc.html

SCG'08, June 9–11, 2008, College Park, Maryland, USA.
ACM 978-1-60558-071-5/08/04.

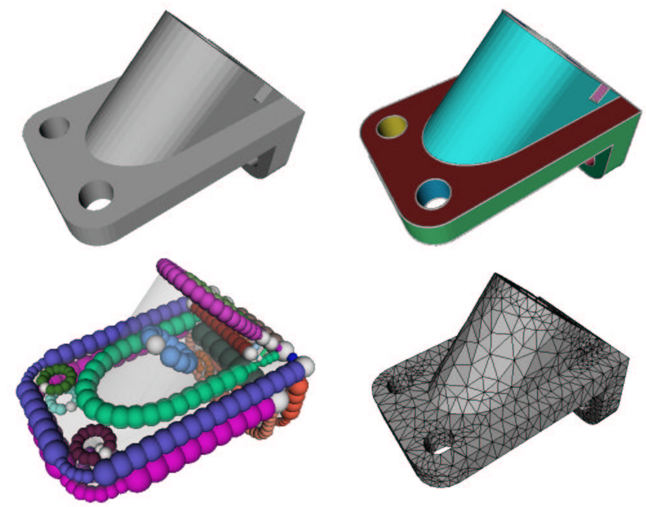

Figure 1: The phases of our algorithm/software. Top left: An input PSC. Top Right: Identification of the non-smooth curves and the smooth patches of the PSC. Bottom Left: Protection of curves by balls (weighted points) drawn as spheres (Protection phase). Bottom Right: Delaunay refinement performed to satisfy topological and geometric criteria (Refinement phase).

The algorithm of [4] has two distinct phases: a *protection* phase and a *refinement* phase. In the protection phase, non-smooth curves and vertices in the input complex are covered with balls. In the refinement phase, these balls are turned into weighted points and a mesh refinement is carried out in the weighted Delaunay triangulation. This algorithm employs some expensive numerical computations for topological and geometric tests that are hard to implement. Cheng, Dey, and Levine [3] eliminated these expensive tests from the refinement phase. In a recent unpublished work [7], we eliminated these tests from the protection phase too. This video shows the working of a software called DelPSC based on this new algorithm.

2. PROTECTION

We protect the input curves and vertices with balls where smooth patches meet giving rise to non-smoothness. A crucial ingredient in the algorithm is that these balls maintain three geometric properties. First, all curves must be covered completely. Second, adjacent balls along a curve must intersect deeply. Third, no three balls should have a common intersection.

Our algorithm places an initial set of balls satisfying the first two properties. It turns out that it is difficult to ensure the third property initially. However, as the balls are refined further in the algorithm, the third property also gets satisfied. Refinement of protecting balls is a main added feature in our new algorithm [7]. Unlike earlier algorithms [4, 3], the protecting balls are not pre-computed and are determined on the fly instead. The main advantage of this approach is that the algorithm does not need to estimate local feature sizes explicitly to determine the placement and sizes of the balls. They are determined on the run by less costly computations.

The video shows the ball refinement process. We say two balls are *adjacent* if they center around two consecutive points ordered along a input curve. We iteratively identify the largest ball intersecting a non-adjacent ball, remove it, and then cover the exposed segment of the curve with a set of shrunken balls. The placements and sizes of the new balls are chosen carefully as described in the full paper [7]. The protecting balls created in this phase are turned into a set of weighted points and inserted into a weighted Delaunay triangulation which becomes the initial triangulation for the refinement phase.

3. REFINEMENT

Our goal is to output the weighted Delaunay triangulation *restricted* to the input, that is, all simplices whose dual Voronoi faces intersect the input domain. In the refinement phase, we insert new points from the domain into the triangulation for topological and geometric violations. For topology we use a disk test as follows. The test requires that the triangles restricted to a manifold patch and incident to a common vertex form a topological disk. If a vertex is incident to multiple patches (if it is a point on an input curve), there should be multiple such disks–one for each patch. Upon failure of the disk test, we either insert a locally furthest point (intersection of Voronoi edges with the input domain) or shrink the largest ball. The choice of which, and its logic, are described in our paper [7].

The video shows the insertion process which eventually fixes the topology. After fixing the topology, the algorithm continues inserting points to satisfy some geometric conditions. Our software has user defined parameters that control the triangle size, aspect ratio, and curvature approximation. We perform refinement by fixing the largest triangle which violates one of these conditions.

We show all triangles which violate any topological or geometric condition in red during the refinement. Correct triangles are shown in gray. As can be observed, some of the gray triangles are not marked to be fixed, though they violate some geometric criterion. This is because fixing them could prevent the refinement from terminating. For example, some triangles have to subtend a small angle and cannot be fixed for good aspect ratio if the input domain contains such small angles to begin with.

4. THE VIDEO

Our software was implemented using CGAL [2], and still frame visualizations of the algorithm were produced using DirectX. Adobe Premiere Pro was used to combine the still frame sequences, produce the audio, and render the final video.

After describing different steps of the algorithm, our video shows some additional examples to illustrate the capability of the software to cope with small input angles, small feature size, boundaries, and non-manifolds. Also, the software has the capability of producing a volume mesh by usual circumcenter insertions for tetrahedra with poor aspect ratio. The software DelPSC can be downloaded from the website [8].

5. ACKNOWLEDGMENTS

We acknowledge the support of NSF, USA under grants CCF-0430735 and CCF-0635008. The weighted Delaunay triangulation code provided by CGAL was essential for the code development.

6. REFERENCES

[1] J.-D. Boissonnat and S. Oudot. Provably good surface sampling and meshing of surfaces. *Graphical Models* **67** (2005), 405–451.

[2] Cgal. Computational Geometry Algorithms Library. http://www.cgal.org.

[3] S.-W. Cheng, T. K. Dey, and J. A. Levine. A practical Delaunay meshing algorithm for a large class of domains. *Proc. 16th Internat. Meshing Roundtable* (2007), 477–494.

[4] S.-W. Cheng, T. K. Dey, and E. A. Ramos. Delaunay refinement for piecewise smooth complexes. *Proc. 18th Ann. ACM-SIAM Sympos. Discrete Algorithms* (2007), 1096–1105.

[5] S.-W. Cheng, T. K. Dey, E. A. Ramos and T. Ray. Sampling and meshing a surface with guaranteed topology and geometry. *Proc. 20th Ann. Sympos. Comput. Geom.* (2004), 280–289.

[6] L. P. Chew. Guaranteed-quality triangular meshes. Report TR-98-983, Comput. Sci. Dept., Cornell Univ., Ithaca, New York, 1989.

[7] T. K. Dey and J. A. Levine. Reducing expensive computations in Delaunay meshing of piecewise smooth complexes. *Manuscript*, 2008.

[8] DelPSC. http://www.cse.ohio-state.edu/∼tamaldey/delpsc.html.

[9] J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Algorithms*, **18** (1995), 548–585.

[10] J. R. Shewchuk. Tetrahedral mesh generation by Delaunay refinement. *Proc. 14th Ann. Sympos. Comput. Geom.* (1998), 86–95.