Probing for Surface Mesh Generation through Delaunay Refinement

Hugo Feree and Pierre Alliez

INRIA

July 31, 2009
<table>
<thead>
<tr>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition of the problem</td>
</tr>
<tr>
<td>Delaunay refinement</td>
</tr>
<tr>
<td>Limitations</td>
</tr>
<tr>
<td>Proposal</td>
</tr>
<tr>
<td>Experimental results</td>
</tr>
</tbody>
</table>
Goal: mesh a smooth closed 2-manifold surface given:

- a convex bounding domain $\Omega$
- an intersection oracle
- a lower bound $\varepsilon > 0$ on the local feature size
- approximation and shape criteria
- nothing more! (especially no initial point set)
Goal: mesh a smooth closed 2-manifold surface given:

- a convex bounding domain \( \Omega \)
- an intersection oracle
- a lower bound \( \varepsilon > 0 \) on the local feature size
- approximation and shape criteria
- nothing more! (especially no initial point set)

with guarantees of:

- topology
- approximation
- shape
- parsimony
Delaunay refinement

Algorithm [Chew, 1993]:
- Compute the Voronoi diagram and the dual 3D Delaunay triangulation of the initial point set
- Compute the Voronoi diagram restricted to the surface
- Check the facets by probing their dual Voronoi edge
- While there are bad facets, refine them
Delaunay refinement

Algorithm [Chew, 1993]:

- Compute the Voronoi diagram and the dual 3D Delaunay triangulation of the initial point set
- Compute the Voronoi diagram restricted to the surface
- Check the facets by probing their dual Voronoi edge
- While there are bad facets, refine them

Many good properties [Boissonnat, Oudot, 2005, 2007]:

- no self intersection
- same topology
- bound and convergence of the
  - Hausdorff distance
  - area
  - normals (the Voronoi edges become orthogonal to the surface)
- shape and approximation quality

But the initialization is left to the user!
Delaunay refinement versus marching cubes

Marching cubes:

▶ careful research
▶ axis dependent (not isotropic)
▶ surface independent, which implies:
  ▶ heavy cost: \( \sim \frac{V}{\varepsilon^3} \) (lots of short probes)
  ▶ oversampling
Delaunay refinement versus marching cubes

Marching cubes:
- careful research
- axis dependent (not isotropic)
- surface independent, which implies:
  - heavy cost: $\sim \frac{V}{\epsilon^3}$ (lots of short probes)
  - oversampling

Delaunay refinement:
- adapts itself to the surface
- axis-independent
- parsimonious
- uses long probes (Delaunay edges) to both refine and discover the surface
- only lacks a careful seeding
Delaunay refinement versus marching cubes

**Figure**: Left: Delaunay refinement (7040 vertices) Right: marching cubes (10440 vertices)
Results with connected surfaces

Figure: Meshes of connected surfaces
Delaunay refinement’s probing ability

**Figure:** All the toruses are meshed after refinement, with only four initial points on the central sphere
Theorem

If $S_0$ is a connected component of diameter $d$ including no other component, at distance at least $d$ of any other connected component and containing at most two points of the triangulation, then $S_0$ will not be meshed by Delaunay refinement.

Figure: Illustration of the theorem in dimension 2
Theorem

If $S_0$ is a connected component of diameter $d$ including no other component, at distance at least $d$ of any other connected component and containing at most two points of the triangulation, then $S_0$ will not be meshed by Delaunay refinement.

Figure: Illustration of the theorem in dimension 2
Theorem

If $S_0$ is a connected component of diameter $d$ including no other component, at distance at least $d$ of any other connected component and containing at most two points of the triangulation, then $S_0$ will not be meshed by Delaunay refinement.

Figure: Illustration of the theorem in dimension 2
Application of the theorem

Figure: The "worst" case
- Probe along the edges of a regular grid: marching cubes
- Probe randomly (done in the surface mesher of CGAL)

**Figure:** Statistics on random spheres with random initial points
• Probe along the edges of a regular grid: marching cubes
• Probe randomly (done in the surface mesher of CGAL)

**Figure:** Cluster of vertices
Dealing with isolated points

Figure: An isolated point
Dealing with isolated points

**Figure:** A persistent facet is made
Dealing with isolated points

Figure: The bad facet is refined
Dealing with isolated points

**Figure:** All the component is meshed
Dealing with isolated points

Figure: Iteration on a grid of spheres
Dealing with isolated points

Figure: Iteration on a grid of spheres
Dealing with isolated points

Figure: Iteration on a grid of spheres
Dealing with isolated points

Figure: Iteration on a grid of spheres

Hugo Feree and Pierre Alliez

Surface mesh generation
Probing the space

Idea: the Voronoi edges are used to refine and discover the surface
⇒ the (largest) Voronoi cells define where the space was not probed

Figure: Illustration in dimension 2: probing a cell
Probing the space

Idea: the Voronoi edges are use to refine and discover the surface ⇒ the (largest) Voronoi cells define where the space was not probed

Figure: Illustration in dimension 2: probing a cell
Probing the space

Idea: the Voronoi edges are use to refine and discover the surface ⇒ the (largest) Voronoi cells define where the space was not probed

Figure: Illustration in dimension 2: probing a cell
Algorithm

\[ Q \leftarrow \emptyset \]
\[ P_0 = \text{probe}(\Omega) \] // Find a first point
\[ \text{initialize\_component}(P_0) \]

\textbf{while} \( Q \neq \emptyset \) \textbf{do}

\textbf{refine}

\textbf{if} \ \exists P \ \text{isolated} \ \textbf{then}

\[ \text{initialize\_component}(P) \] // Deal with isolated points

\textbf{else}

\textbf{repeat}

\[ C = \text{pop} Q \]
\[ \text{probe}(C) \] // Probe a cell

\textbf{until} intersection\_found \textbf{or} \( Q = \emptyset \)

\textbf{end if}

\textbf{end while}
Examples

Figure: Simple Delaunay refinement (blue) and after isolated points treatment (red)
Statistics with random spheres

Figure: Statistics for mesh generation with isolated points treatment
An adaptive method

This algorithm can be accelerated knowing (optional) informations:

- some initial points
- the number of connected components
- a maximum inside radius $R > \varepsilon$

**Figure:** $R$ can be arbitrarily large compared with $\varepsilon$ and its knowledge can avoid many probing steps: illustration in dimension 2
Conclusion

We have provided:

▶ a generic adaptative meshing algorithm based on Delaunay refinement
  ▶ with the same quality and parsimony
  ▶ but without asking for a user’s initialization.

▶ a partial implementation with concluding results.

Rest to do:

▶ an implementation of the whole algorithm
▶ an accurate study of complexity in time and in number of vertices
Thanks!

Questions?

Hugo Feree and Pierre Alliez

Surface mesh generation