Steiner Point Reduction in Planar Delaunay Meshes

Ahmed Abdelkader*, Scott A. Mitchell†, Mohamed S. Ebeida†

* Department of Computer Science, University of Maryland
† Sandia National Laboratories
Outline

- Background
- Motivation
- Proposed Method
- Sample Results
- Discussion
- Extras
Background: Delaunay Triangulation

- Empty circumcircles
- Maximizes minimum angle
- Unique if points in general position

(Left) A Deluanay Triangulation. (Right) Corresponding Voronoi Diagram (constant weighted dual).
[Source: Wikipedia]
Background: Constrained DT

- Empty circumcircles (Not true Delaunay)
- Fewer elements
- Better bounds (min. angles + grading)

(Left) A Delaunay Triangulation with empty circumcircles. (Right) Circumcircle of $t_i$ contains point $p$, which is not to the interior of $t_i$. [Source: Wikipedia (left), Chrisochoides et al. (right)]
Background: Meshing Algorithms

- **Input:** \((P, [S]), \alpha\). **Output:** good [conformal] Delaunay complex.

- **Ruppert's Delaunay Refinement Algorithm (DR) [1]**
  - “... perhaps the first theoretically guaranteed meshing algorithm to be truly satisfactory in practice” [2].
  - Halts for an angle constraint of up to \(20.7^\circ [1], 26.45^\circ [3]\).

- **Chew’s Second Algorithm [4]**
  - Terminates with minimum angle up to \(26.57^\circ [2], 28.6^\circ [5]\).

---

Background: Why Triangulate?

- Complicated object -> collection of simple objects

- Interpolation
  - Graphics rendering

- Finite element analysis
  - Convergence and accuracy
  - Condition number of the linear system
Motivation

- Reduce mesh size (number of points)
- Retain angle bounds
- Preserve all features
Proposed Method: Sifting

1. Replace an edge (2 points) with 1 point

2. Constrain the region of valid replacement points

3. Sample uniformly from this region

4. Repeat until no more sifting is possible
Constraint #1: Neighboring Circumcircles
Constraint #2: No Thin Triangles
Constraint #2: No Thin Triangles (Cont.)
Examples
Constraint #3: Boundary Segments
# Quantitative Results

<table>
<thead>
<tr>
<th>model_name</th>
<th>#input points</th>
<th>(#Triangle’s output size, reduction ratio)</th>
</tr>
</thead>
<tbody>
<tr>
<td>./triangle -q</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>B5</td>
<td>71</td>
<td>139</td>
</tr>
<tr>
<td>Spiky</td>
<td>229</td>
<td>330</td>
</tr>
<tr>
<td>Dolphin</td>
<td>260</td>
<td>471</td>
</tr>
</tbody>
</table>

model_name | #input points | (#Triangle’s output size, reduction ratio)
Sensational Results: B5 Model

(a) B5 (71)  (b) Triangle (326)  (c) Sifted (43%)
Sensational Results: Spiky Model

(d) Spiky (229)  
(e) Triangle (715)  
(f) Sifted (56%)
Sensational Results: Dolphin Model

(g) Dolphin (260)  (h) Triangle (3409)  (i) Sifted (78%)
Discussion

● Sifting order
  ○ Edges chosen at random

● Random Sampling
  ○ Replacement points chosen at random (from region)

● Runtime
  ○ Expected number of “sifting attempts” per edge

● Quantify improvement?
Questions?

Thank You!

akader@cs.umd.edu
Extra Slide #1: Adaptive Local Max.

Varying the angle bound used.
Left to right: 20, 30, 40, 50.
Extra Slide #2: Surface Teaser

~1: Neighboring Circumspheres

~2: No Thin Triangular Faces
Goal: reduce the number of points while retaining the angle bounds.

Local update strategy (Sifting)

- Remove 2 points
- Constrain sampling region
  - Neighbor Circumcircles
  - Angle bounds
- Pick a replacement point

Example: 78% reduction