On Computing Meshes with Large Smallest Angles

Hale Erten Alper Üngör

Dept. of Computer & Information Science & Engineering, University of Florida
{herten,ungor}@cise.ufl.edu

Abstract. We present a variant of the Delaunay refinement, method which inherits the theoretical guarantees of the original Delaunay refinement algorithm. For a given input domain (a set of points or a planar straight line graph), and a threshold angle α, the Delaunay refinement method computes triangulations (meshes) that have all angles at least α. For various input domain types, Delaunay refinement algorithm is proven to terminate with correct output for $\alpha \leq 30°$. In practice, the original Delaunay refinement algorithm works for values of α as high as 35°. The new variant of the refinement algorithm terminates for larger values of α, as high as 38°.

1 Introduction

We consider the following two-dimensional geometric optimization problem: *Compute the smallest size triangulation of a given domain (collection of points and/or segments) such that all the triangles in the triangulation are of good quality.* Quality constraint is motivated by the numerical methods used in engineering applications. A triangle is said to be good if its smallest angle is bounded from below, or equivalently, its radius-edge ratio (circumradius over shortest edge length) is bounded from above. A lower bound of α on the smallest angle translates to an upper bound of $\beta = (2 \sin \alpha)^{-1}$ on the radius-edge ratio. A number of (approximation) solutions has been suggested for this problem. Quadtree based methods [1] and Delaunay based methods [2, 5, 6] provide roughly the same theoretical guarantees. However, Delaunay based methods computes triangulations with smaller number of triangles.

Delaunay refinement method involves first computing an initial Delaunay triangulation of the input domain, and then iteratively adding points called *Steiner points* to improve the quality of the triangulation. Traditionally, circumcenters of bad triangles are used as Steiner points [2, 5]. We recently introduced a new type of Steiner points, called *off-centers*, as an alternative to circumcenters [6]. This lead to the design of the first time-optimal Delanuay refinement algorithm [4]. In practice, off-center insertion algorithm results in significant reduction in the output size [6]. It is now used in the popular Delaunay refinement software `Triangle (version 1.6)` [1]. Both the original circumcenter and the off-center insertion variants of the Delaunay refinement suffers from a *termination problem* for large values of α. In this note, we explore ideas to alleviate this problem.

2 Termination Problem

Termination guarantee of the Delaunay refinement algorithms relies on a packing argument. This argument applies for small values of α, (i.e. $\alpha \leq 30°$) where circumcenter insertion does not gradually decrease the shortest pairwise vertex distance. For $\alpha > 30°$, however, the iterative refinement process could introduce smaller and smaller features and may not terminate. In practice, such phenomenon starts occurring for $\alpha > 34°$, regardless of the type of input data. See Figure 1 (a-d) to see how `Triangle 1.6` suffers from this problem. It is interesting to note that the problem starts occurring in regions far away from the boundary and the input features.

3 Staying Away from Existing Vertices

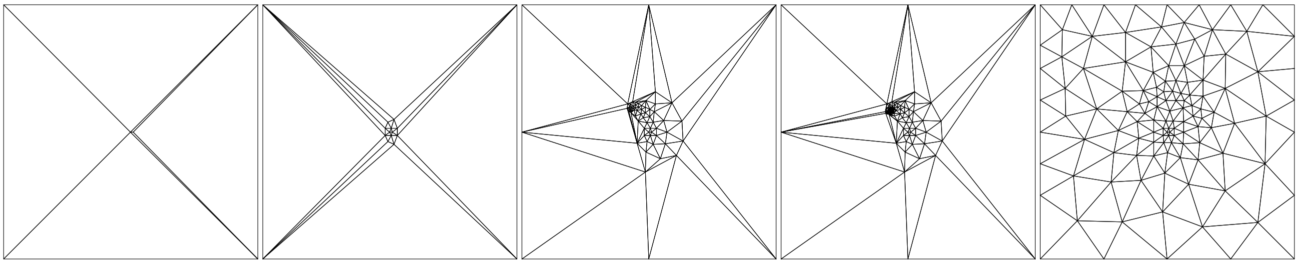
Let pqr be a bad triangle in a triangulation and pq be its shortest edge. Consider the circle that goes through pq and of radius $\beta|pq|$ whose center is on the same side of pq as r. The disk bounded by this circle, which we call the *petal* of pq must include another vertex. Otherwise, there would be a bad triangle in the triangulation incident to p and q. In the following algorithm, we suggest to pick a Steiner point inside the petal furthest away from all existing vertices.

Algorithm 1

Compute the Delaunay triangulation of the input
while $\exists$ a bad triangle pqr with shortest edge pq
  Insert a point $x \in petal(pq)$ which is furthest from all existing vertices

Note that this point is either a Voronoi vertex (but not necessarily the circumcenter of pqr) or on a Voronoi edge. For bad triangles with very large radius-edge ratio (i.e., circumcenter of pqr is outside the petal of pq), the intersection of the bisector of

[1] `http://www-2.cs.cmu.edu/~quake/triangle.html`



(a) (b) (c) (d) (e)

Figure 1: Illustration of the termination problem. Input is a pair of points at unit distance from each other inside a box of side length 100 units. Delaunay triangulation of the input is shown in (a). For $\alpha = 36°$, the previous Delaunay refinement algorithms do not terminate. We interrupted the execution of `Triangle 1.6` after the insertion of (b) 10 Steiner points (c) 100 Steiner points (d) 1000 Steiner points. Output of the new algorithm for $\alpha = 38°$ is shown in (e).

pq and the boundary of the petal of pq is used as such a Steiner point [6]. For bad triangles with relatively small radius-edge ratio (i.e., circumcenter of $pqr \in petal(pq)$), such a Steiner point could be found by searching a local neighborhood on the Voronoi diagram. In this case, x could be the circumcenter of pqr or the circumcenter of a nearby triangle (Figure 2 (left)) or a point on a Voronoi edge (Figure 2 (right)).

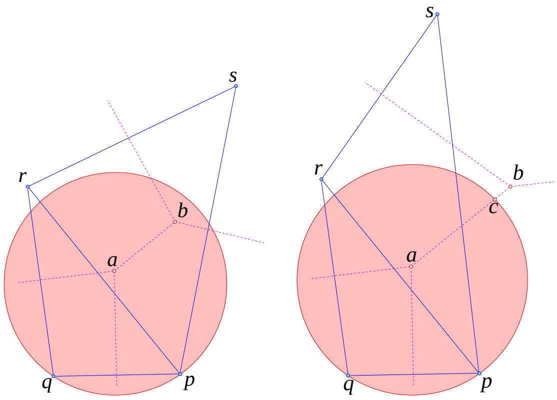


Figure 2: Find a point inside the shaded disk of pq that is furthest away from all existing vertices. Such a point can be a nearby Voronoi vertex (labeled as b (left)); or a point on a Voronoi edge (labeled as c (right).

Our approach is similar to sink insertion [3] in maximizing the distance of the Steiner points from the existing vertices. The main difference is we limit the choice to a local neighborhood of bad triangles.

We implemented this idea by modifying `Triangle 1.6` and run experiments on various data sets and point distributions. We observed that the practical termination bound of Delaunay refinement is improved by about 2° on average. Figure 3 plots the performance of the new algorithm compared with `Triangle 1.6`. The output mesh of the new algorithm is shown in Figure 1 (e). While the previous algorithm gets into the termination problem for $\alpha \geq 35°$, the new algorithm terminates with for $\alpha = 38°$. Performance plots are similar for various data sets.

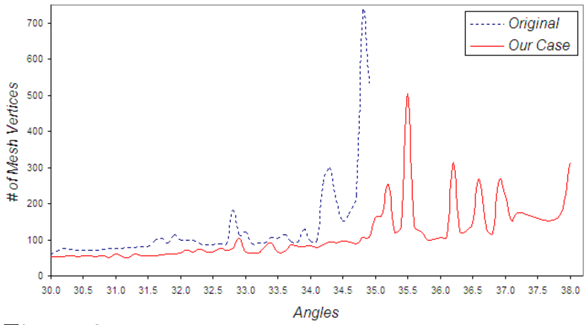


Figure 3: Plot of the number of output mesh vertices vs. the threshold angle α.

4 Discussions

Note that the termination and size complexity bounds given for the previous Delaunay refinement apply here for small values of α, as we are more cautious in introducing short features. It would be interesting if we can prove the same theoretical bounds for $\alpha > 30°$. It would be also interesting to further improve the practical performance angle bounds say for $\alpha \geq 45°$. This would imply non-obtuse or acute angle triangulations.

References

[1] M. Bern, D. Eppstein, and J. Gilbert. Provably good mesh generation. *J. Comp. System Sciences* 48:384–409, 1994.

[2] L.P. Chew. Guaranteed-quality triangular meshes. TR-89-983, Cornell Univ., 1989.

[3] H. Edelsbrunner and D. Guoy. Sink insertion for mesh improvement. *Proc. ACM Symp. Comp. Geometry*, 115–123, 2001.

[4] S. Har-Peled and A. Üngör. A time-optimal Delaunay refinement algorithm in two dimensions. *Proc. ACM Symp. Comp. Geometry*, 228–236,2005.

[5] J. Ruppert. A new and simple algorithm for quality 2-dimensional mesh generation. *Proc. 4th ACM-SIAM Symp. on Disc. Algorithms*, 83–92, 1993.

[6] A. Üngör. Off-centers: A new type of Steiner points for computing size-optimal quality-guaranteed Delaunay triangulations. *Proc. LATIN*, LNCS 2976, 152–161, 2004.