Delaunay Edge Refinements

Alan Saalfeld
Bureau of the Census
email: saalfeld@cs.umd.edu

Abstract

A triangulation refinement of a triangulation T is a triangulation T', such that every triangle of T' lies entirely in a triangle of T. We show that every triangulation has a triangulation refinement that is Delaunay.

Constrained Delaunay triangulations currently provide a useful tool for generating elevation maps over triangulated terrain. Critical edges representing ridges or valleys are "forced" to be part of the triangulation; and the remaining vertices are triangulated using a modified empty circle Delaunay condition [CHEW]. One problem with using the constrained Delaunay triangulation for cartographic applications is that the forced edges produce narrow triangles with poor numeric properties.

An alternative approach to using a constrained Delaunay triangulation to control large critical edges is to chop up the critical edges into short enough segments (by adding additional vertices along the critical edges) so that the constrained Delaunay triangulation of all of the short segments is nothing other than the true Delaunay triangulation of the augmented vertex set. We show how to subdivide the constraint edges efficiently to guarantee that the resulting segment pieces will be edges of a Delaunay triangulation.

1 Triangulations and Refinements

This paper deals only with triangulating finite point sets in the plane and planar triangulations. A triangulation of a planar point set is a maximal straight-line plane graph whose vertex set is exactly the point set.

A *triangulation refinement* of a triangulation T is a triangulation T', such that every triangle of T' lies entirely in a triangle of T. A necessary, but not sufficient condition that T' be a triangulation refinement of T is that the vertex point set of T be a subset of the vertex point set of T'.

The following lemma provides a simple characterization of triangulation refinements:

Lemma 1 *Suppose that T and T' are triangulations covering the same region. Then T' is a triangulation refinement of T if and only if every edge of T is a union of edges of T'.*

If the vertices of a triangulation refinement T' of T all lie on the edges of T, then we will call T' a *triangulation-edge refinement* of T.

For some applications, triangulation refinements may require the addition of vertices on the interior of triangles of the original triangulation. We will see that refining a triangulation to make

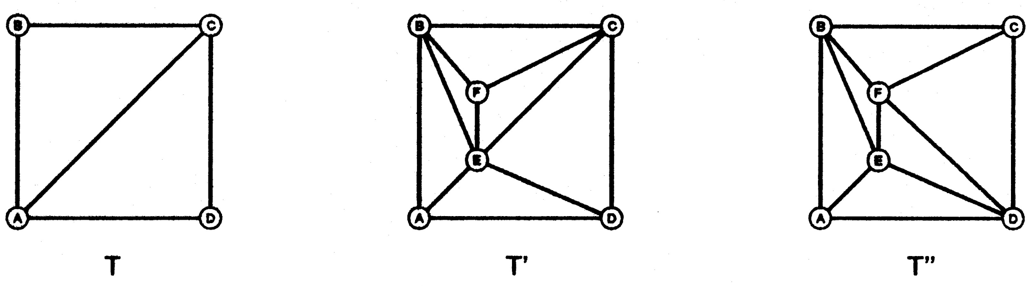

Figure 1: T' is a triangulation refinement of T; T'' is not.

the refinement Delaunay does *not* require adding interior vertices. We will use the previous lemma and a characterization of the edges of a Delaunay triangulation to show that every triangulation has a triangulation-edge refinement that is Delaunay.

2 Delaunay Conditions

The well-known empty circle condition for triangles of a Delaunay triangulation has an immediate characterization corollary for edges of Delaunay triangulation:

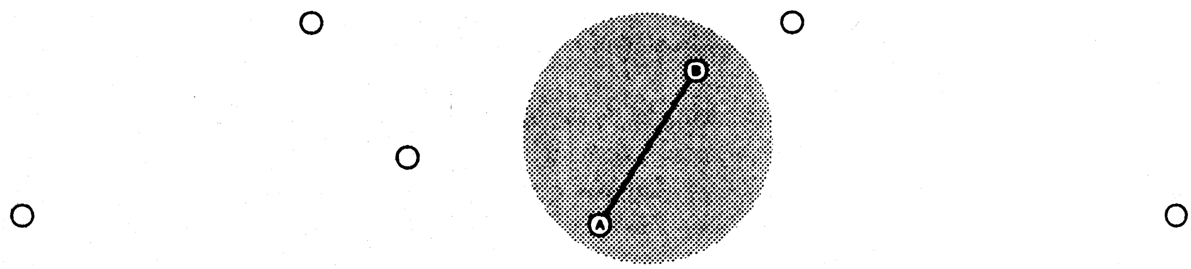

Figure 2: A Delaunay disk verifying the edge condition for AB.

Lemma 2 (Edge Condition) *The segment joining two vertices is an edge in a Delaunay triangulation if and only if there exists some closed disk containing the two vertices and containing no other vertices in its interior.*

Note that the edge condition is one of existence—we only have to find one such disk to guarantee that an edge belongs to a Delaunay triangulation. We will call any such verifying disk a Delaunay disk for the edge. We will use the edge condition extensively on small edges.

3 Constrained Triangulations

A constrained triangulation is a triangulation, some or all of whose edges have been prespecified. One may regard the constraints as a plane line graph and the constrained triangulation as a maximal plane extension straight line graph.

A constrained Delaunay triangulation is a triangulation that is "as Delaunay as possible" subject to the condition that it contain the prespecified edges. We refer the reader to [CHEW] for a more formal definition of constrained Delaunay triangulations and details on their properties. In cartographic applications such as elevation mapping, constraining edges allows one to guarantee that a function such as elevation is properly defined along the edge.

4 Edge Refinements

Given a collection of vertices and some set of edges in a plane line graph, we may subdivide the edges into segments by adding additional vertices along the edges and splitting the edges at the new vertices. If ***after*** such edge splitting, we ***then*** build a triangulation, we call the resulting triangulation a constrained edge refinement. Such a resulting triangulation need not be a triangulation refinement of a constrained triangulation resulting from the initial edge constraint set.

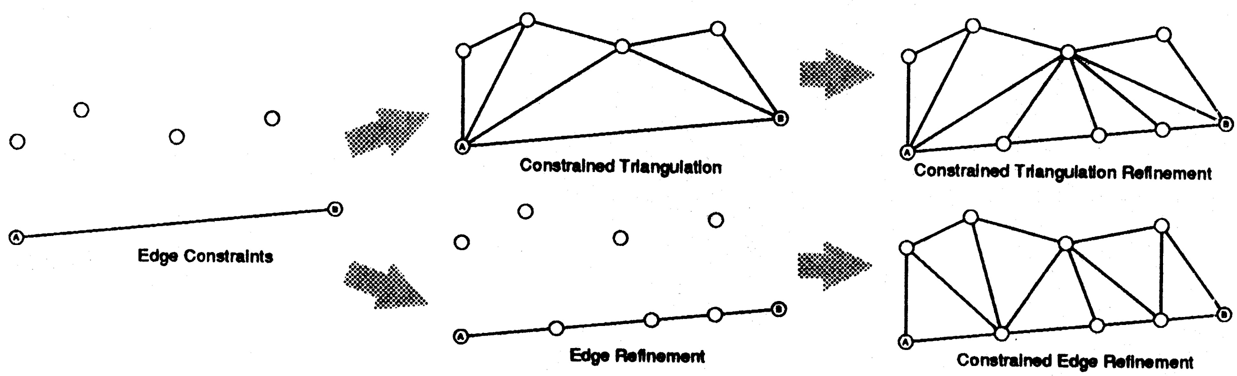

Figure 3: A constrained edge refinement that is not a triangulation refinement.

5 Existence Theorem

Theorem 1 ***Every triangulation has a Delaunay triangulation-edge refinement.***

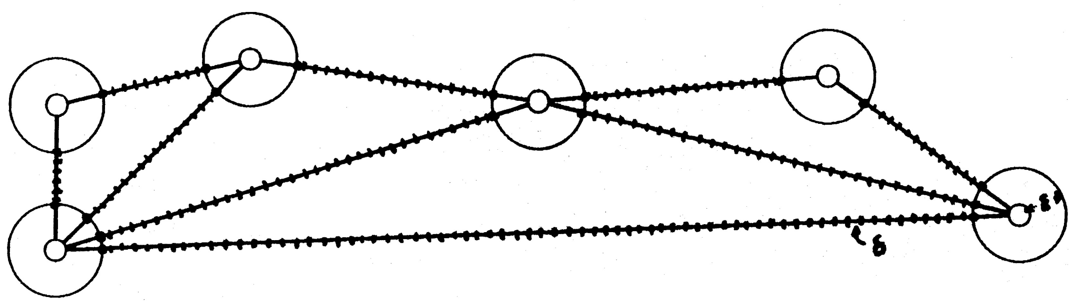

Figure 4: Subdividing triangulation edges into Delaunay-sized pieces.

The proof idea is very simple: we chop up the edges of the given triangulation in such a way that we can guarantee that each edge piece has a verifying Delaunay disk. First we chop up the edges (i.e. add vertices) near the original triangulation vertices; then we subdivide the remaining edge pieces. The following approach works: Let ϵ be the minimum edge length in the initial triangulation. Around each vertex of the original triangulation, draw a circle of radius $\epsilon/3$. Add all of the intersections of the circles and triangulation edges to the vertex set. Let δ be the minimum distance between vertices in the augmented vertex set. Finally subdivide the middles of each original triangulation edge into pieces of size less than δ. Verifying that every edge piece now has a Delaunay disk is straightforward.

6 Delaunay Edge Refinement Algorithm

If we are just given a vertex set and some constrained edges we may also subdivide the constraint edges (without having a full constrained triangulation) in a very natural iterative fashion:

Theorem 2 *Every plane graph has a Delaunay edge refinement that can be built by iterated intersection of the graph edges with a Delaunay triangulation of a growing vertex set until such iteration stabilizes, and the vertex set stops growing.*

We sketch the actual algorithm below:

```
Input:  A plane graph G = G(V,E), consisting of
                      V, a vertex set (with coordinates), and
                      E, a set of (constrained) edges.

Algorithm:

  Initialize:  Set V', the augmented vertex set, equal to V.

  Loop:  1. Build the ordinary Delaunay triangulation T on V'.
         2. Overlay G on T, computing the set P of segment intersection points.
         3. If P is not contained in V', Then add P to V' and return to (1).

                                          Else output V' and T; and STOP.
```

The proof that the algorithm terminates and produces a Delaunay edge refinement is again a straightforward application of the Delaunay disk test. One important observation that has practical significance for the elevation mapping application mentioned earlier is that if the edge constraints are isolated (by which we mean that no two distinct edges of E properly intersect the same triangle in a Delaunay triangulation of the vertices of V), then the algorithm terminates on the second execution of the loop; and the set V' is updated only once.

7 References

CHEW, L. Paul, 1987, *Constrained Delaunay Triangulations*, Proceedings of the Third Annual Symposium on Computational Geometry, 215-222.