Constrained Higher Order Delaunay Triangulations*

Joachim Gudmundsson Herman Haverkort Marc van Kreveld

Dept. of Computer Science, Utrecht University,
P.O.Box 80.089, 3508 TB Utrecht, the Netherlands.
joachim@cs.uu.nl, herman@cs.uu.nl, marc@cs.uu.nl

1 Introduction

A previous paper by Gudmundsson et al. [3] studied a new type of triangulation called *higher-order Delaunay triangulation.* It is a class of well-shaped triangulations for a given point set. Such triangulations are useful in realistic terrain modeling on a set of points in the plane with known elevation. Often, in terrain modeling it is desirable to force a given set of edges to be part of the triangulation. These edges can come from contour lines or from the drainage network [2, 4, 6]. Motivated by this, we study *constrained higher-order Delaunay triangulations* in this paper. We first repeat the definition of higher-order Delaunay triangulations:

Definition 1 *A triangulation of a set P of points is an* order-k Delaunay triangulation *if for any triangle of the triangulation, the circumcircle of that triangle contains at most k points of P.*

So a normal Delaunay triangulation is an order-0 Delaunay triangulation, and for any positive integer k, there can be many different order-k Delaunay triangulations. By definition, any order-k Delaunay triangulation is also an order-k' Delaunay triangulation if $k' > k$.

Another important concept from Gudmundsson et al. [3] is the *useful order* of an edge:

Definition 2 *For a set P of points, the* order *of an edge between two points $p, q \in P$ is the minimum number of points inside any circle that passes through p and q. The* useful order *of an edge is the lowest order of a triangulation that includes that edge.*

In this paper we study constrained higher-order Delaunay triangulations, which must include a given set of edges in the triangulation. Note that the order of a Delaunay triangulation with only one constraining edge is exactly the useful order of that edge. This paper studies the case of more than one constraining edges. We study the following questions:

1. Given a triangulation T (all edges are constraining), determine its order.
2. Given a set P of n points and a set E of edges, determine the lowest order Delaunay triangulation of P that includes the edges of E.

The first question we can solve in two ways. Circular range counting gives an efficient algorithm for large orders, and higher-order Voronoi diagrams are the basis of an efficient algorithm for lower orders.

The main result we have for the second question is that if every edge in E has useful order k or less, then a triangulation of E and P exists that has order at most $2k - 1$. In fact, this triangulation is the constrained Delaunay triangulation. The bound is worst-case optimal: there are point sets with constraining edges, all of useful order k or less, for which any triangulation has order at least $2k - 1$.

Throughout this paper we assume general position, that is, no three points of a point set P lie on a line, and no four points of P lie on one circle.

*J.G. is supported by the Swedish Foundation for International Cooperation in Research and Higher Education. H.H. is supported by the Netherlands Organization for Scientific Research (NWO).

2 Determining the order of a triangulation

Given a triangulation T, we can determine its order k in one of two ways, based on the observations and algorithms given before in [3]. The first is efficient for any k, in particular, it is the best we can do if the unknown value k is at least $\sqrt{n}$ with some logarithmic factors. The second algorithm is more efficient when k is constant or a function that grows slower than $\sqrt{n}$ with logarithmic factors. Small values of k are expected to be most important in practical situations.

Both algorithms begin by determining the $O(n)$ circles through the three points of any triangle in the triangulation. Then we find out how many points lie in these circles. The circle containing the largest number of points determines the order of the triangulation.

The first algorithm is based on a circular range searching data structure on P that can answer point counting queries for query circles efficiently. For various storage requirements m, a data structure of space $O(m)$ exists that answers such circular range counting queries in $O(n/m^{1/3}\log(m/n))$ time [1]. The structure takes $O(m\log^{O(1)} m)$ time to construct. We choose m to be $n^{3/2}$. A triangulation gives rise to $O(n)$ circular range queries; the maximum count returned yields the order of the triangulation. So this solution takes $O(n^{3/2}\log^{O(1)} n)$ time in total.

The second solution comes down to choosing a value k' and testing whether the actual order k is less than k' or not. This can be done by computing the k'-th order VD and preprocessing it for point location queries. A query returns the k'-th closest point. To find out — for a query circle — whether it contains less than k' points, we query with the center of the circle and find the k'-th closest point, which is tested explicitly for containment in the circle. If for all $O(n)$ query circles the k'-th closest point lies outside, we know that the order is less than k'.

The k'-th order VD can be computed and preprocessed for planar point location in $O(nk'\log n)$ time [5]. We start with $k' = 1$, and if k appears to be larger, we double k' and test again. After at most $O(\log k)$ attempts, we find an interval of values $[2^i, 2^{i+1}]$ that must contain k. By binary search on this interval, we take another $O(\log k)$ steps to determine the exact order of the triangulation T. So in total, this method takes $O(nk\log n\log k)$ time.

Theorem 1 *Given a triangulation with n vertices, its order k can be determined in $O(n^{3/2}\log^{O(1)} n)$ time and in $O(nk\log n\log k)$ time.*

3 Completing to a Higher Order Delaunay Triangulation

Assume that a set P of n points and a set E of edges are given. P must include the endpoints from E. This section deals with computing a triangulation of P that includes the edges of E. We would like the triangulation to have the lowest possible order.

As mentioned in the introduction, a previous paper [3] includes the case $|E| = 1$. In case there is only one constraining edge $\overline{uv}$, we can determine the lowest k for which $\overline{uv}$ is a useful order-k Delaunay edge. Then we can complete it to a triangulation only using triangles whose circumcircle contains no more than k points. One of the triangles incident to $\overline{uv}$ has order k, or both, and no other triangle needs to have higher order. In the completion, $\overline{uv}$ will be part of triangles $\triangle uvs$ and $\triangle uvt$. Points s and t are the first points hit by a circle squeezed in between u and v from the one side and from the other side, see Figure 1(a).

The case with more constraining edges is more difficult than the case of one constraining edge. In [3] it was shown that if all edges of E are Delaunay or useful first order Delaunay, then a completion to a first order Delaunay triangulation exists and can be computed in $O(n\log n)$ time. It is simply the constrained Delaunay triangulation. But as soon as E contains edges that are useful k-order with $k > 1$, we cannot necessarily complete it to an order-k Delaunay triangulation anymore, as shown in the next theorem.

Theorem 2 *Let P be a set of points and let E be a set of edges that are all useful order-k Delaunay edges, with $k \geq 2$. Then we have:*

(i) For any sets P and E, the constrained Delaunay triangulation has order $2k-2$.

(ii) For some sets P and E, any constrained triangulation has order at least $2k-2$.

(iii) For some sets P and E, the constrained Delaunay triangulation does not have order smaller than $2k-2$, but some other constrained triangulation has order k.

Proof: We begin with (ii), which is shown by example. Figure 1(b) excluding point s shows a point set with 9 points and 2 constraining edges. Any constrained triangulation must contain $\triangle uvw$, and hence the number of points in the grey circle determines the order. The four other circles show the useful order of the two constraining edges, which is 4. This example immediately generalizes to having $k-1$ points in each of the two circle parts left of $\vec{uv}$ and right of $\vec{uw}$. Then the edges $\overline{uv}$ and $\overline{uw}$ have useful order k, and the circle through u, v, w contains $2k-2$ points inside.

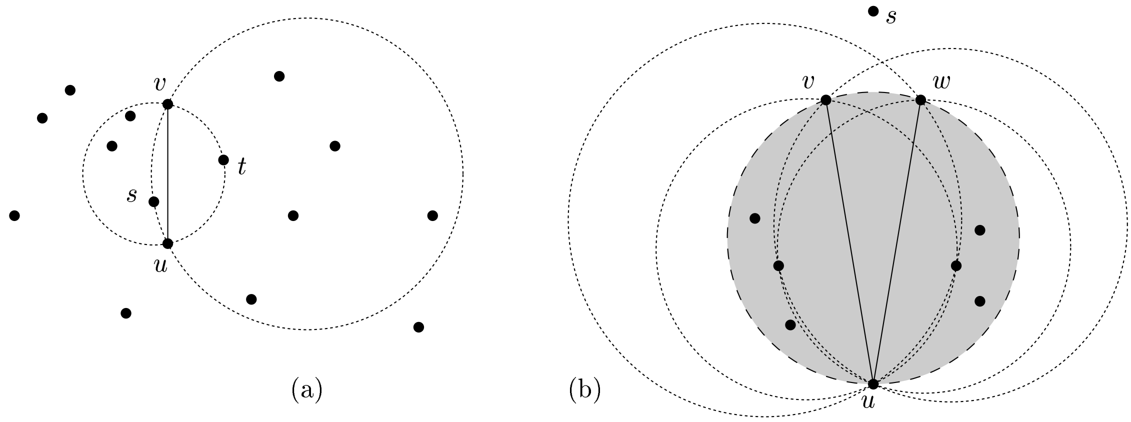

Figure 1: (a) Illustration of the first-points-hit (s and t). (b) Illustration of the proof.

Part (iii) of the lemma also follows from Figure 1(b), now including point s. The constrained Delaunay triangulation has order $2k-2$, but flipping the edge $\overline{vw}$ to $\overline{us}$ reduces the order to k.

For part (i), consider the constrained Delaunay triangulation of P and E, and any triangle u, v, w of it. The circle through u, v, w can only contain points that are 'behind' edges of the CDT, see Figure 2. These edges must be constraining edges of E. (More correctly: for any point $p \in P$ inside the circle $C(u,v,w)$ there must be a constraining edge intersecting $C(u,v,w)$ twice and which has $\triangle uvw$ and point p on different sides.) Let $E' \subseteq E$ be the constraining edges that intersect $C(u,v,w)$ twice, separate a point of P inside $C(u,v,w)$ from $\triangle uvw$, and are closest to $\triangle uvw$ among these (that is, no other constraining edge lies in between: in Figure 2, the dashed edge is not in E').

If there is only one edge $e \in E'$, there can be at most k points behind it inside $C(u,v,w)$ because the first-point-hit for the edge e in the direction of $\triangle uvw$ will be point u, v, or w, or some point hit even before. That will give a circle with those same points inside. Since this circle is one of the two that determine the useful order of e, there can be at most k points inside. Hence, $C(u,v,w)$ can contain at most k points as well.

If E' contains at least two edges, consider any two of them, say e_1 and e_2. Let C_1 and C_2 be the circles through the endpoints of e_1 and e_2 and the first-point-hit behind the edges e_1 and e_2, respectively, see Figure 2. These two circles together cover the whole of $C(u,v,w)$. Since these circles are also the ones that determine the useful order of the constraining edges e_1 and e_2, which is at most k, the circles C_1 and C_2 can contain at most k points each. These include the points u, v, w, unless the endpoints of e_1 (or e_2) happen to be u, v, or w. But both C_1 and C_2 contain at least one of u, v, w. Hence, at most $k-1$ other points of P can lie inside each. It follows that at most $2k-2$ points of P can lie inside $C(u,v,w)$, which shows that the order of $\triangle uvw$ is at most $2k-2$. Since this triangle was any triangle of the CDT, the part (i) of the lemma follows. $\square$

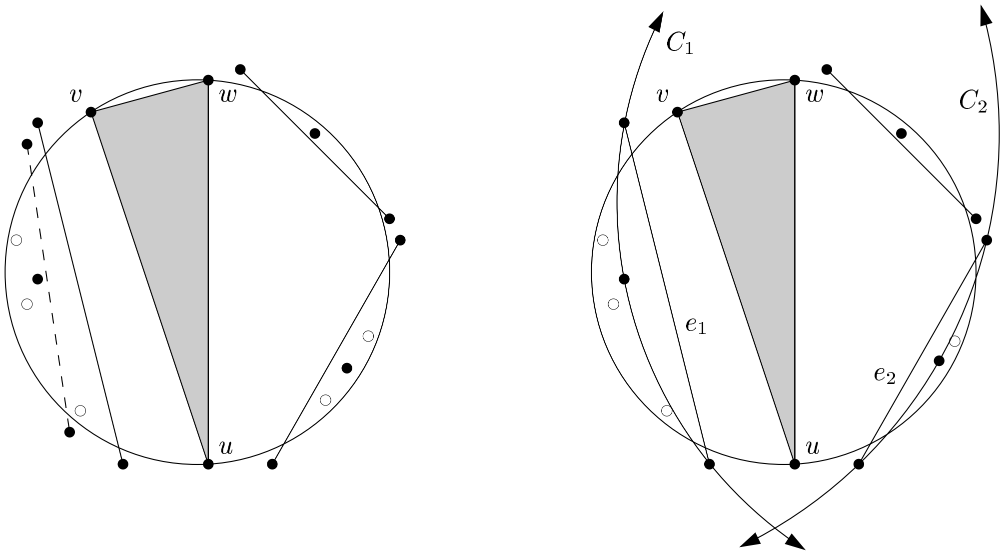

Figure 2: The order of a triangle in a constrained Delaunay triangulation.

4 Conclusions

We have extended results on higher-order Delaunay triangulations and generalized them to constrained higher-order Delaunay triangulations. The application of constrained higher order Delaunay triangulations lies in realistic terrain modeling, where a known river network gives the set of constraining edges. The next research issue is to integrate other criteria for realistic terrain modeling [6] by optimizing over the constrained higher-order Delaunay triangulations.

An open problem that arises in this paper is the computation of the lowest order completion of a set of useful order-k Delaunay edges to a triangulation. The constrained Delaunay triangulation only gives a 2-approximation of the lowest order.

References

[1] P. K. Agarwal. Range searching. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 31, pages 575–598. CRC Press LLC, Boca Raton, FL, 1997.

[2] L. De Floriani and E. Puppo. A survey of constrained Delaunay triangulation algorithms for surface representaion. In G. G. Pieroni, editor, *Issues on Machine Vision*, pages 95–104. Springer-Verlag, New York, NY, 1989.

[3] J. Gudmundsson, M. Hammar, and M. van Kreveld. Higher order Delaunay triangulations. *Comput. Geom. Theory Appl.*, 23:85–98, 2002.

[4] M. McAllister and J. Snoeyink. Extracting consistent watersheds from digital river and elevation data. In *Proc. ASPRS/ACSM Annu. Conf.*, 1999.

[5] E.A. Ramos. On range reporting, ray shooting and k-level construction. In *Proc. 15th Annu. ACM Symp. on Computational Geometry*, pages 390–399, 1999.

[6] B. Schneider. Geomorphologically sound reconstruction of digital terrain surfaces from contours. In T.K. Poiker and N. Chrisman, editors, *Proc. 8th Int. Symp. on Spatial Data Handling*, pages 657–667, 1998.