Constrained Higher Order Delaunay Triangulations

Joachim Gudmundsson

Herman J. Haverkort

Marc van Kreveld

institute of information and computing sciences, utrecht university

technical report UU-CS-2004-007

www.cs.uu.nl

Constrained Higher Order Delaunay Triangulations*

Joachim Gudmundsson[1] Herman J. Haverkort[2] Marc van Kreveld[2]

[1] Dept. of Mathematics and Computing Science, TU Eindhoven,
5600 MB Eindhoven, the Netherlands.

[2] Dept. of Computer Science, Utrecht University,
P.O.Box 80.089, 3508 TB Utrecht, the Netherlands.

h.j.gudmundsson@tue.nl, herman@cs.uu.nl, marc@cs.uu.nl

Abstract

We extend the notion of higher-order Delaunay triangulations to *constrained higher-order Delaunay triangulations* and provide various results. We can determine the order k of a given triangulation in $O(\min(nk \log n \log k, n^{3/2} \log^{O(1)} n))$ time. We show that the completion of a set of useful order-k Delaunay edges may have order $2k-2$, which is worst-case optimal. We give an algorithm for the lowest-order completion for a set of useful order-k Delaunay edges when $k \leq 3$. For higher orders the problem is open.

1 Introduction

A previous paper by Gudmundsson et al. [7] 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 [4, 8, 10]. 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 standard 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. [7] 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.*

*J.G. and H.H. are supported by the Netherlands Organization for Scientific Research (NWO).

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 edge. 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 non-intersecting edges between points of P, 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 P and E exists that has order at most $2k-2$. 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 $2k-2$. Furthermore, we show that for a set of useful order-k Delaunay edges with $k \leq 3$, the lowest-order completion can be computed in $O(n \log n)$ time. We do not have any polynomial-time algorithm for $k \geq 4$.

Constrained Delaunay triangulations have been studied extensively before. Chew [3] was the first to give an optimal, $O(n \log n)$ time algorithm to compute them. Other related research on triangulations is done by Devillers et al. [5], who analyze for a given triangulation T what the minimal set of edges of T is, such that T can be reconstructed from these edges using a constrained Delaunay triangulation. Bern et al. [2] give algorithms that optimize other quality criteria than the Delaunay criterion. Dyn et al. [6] consider data dependent triangulations for surface fitting; several other papers on this topic exist.

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 a circle.

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 [7]. 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 Voronoi Diagram 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 Voronoi Diagram can be computed and preprocessed for planar point location in $O(nk' \log n)$ time [9]. 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 time $O(\min(n^{3/2} \log^{O(1)} n,\, nk \log n \log k))$.*

3 Completing to a lowest order Delaunay Triangulation

Assume that a set P of n points and a set E of non-intersecting edges are given, where P includes the endpoints from E. Edges of E may share endpoints, however. 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 [7] 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, except for useful order-1 Delaunay edges. In [7] it was shown that if all edges of E are Delaunay or useful order-1 Delaunay, then a completion to an order-1 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 order-k 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 with the points of P as endpoints, such that any two edges of E are disjoint, or only intersect at a shared endpoint. If all edges of E are 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 at most $2k-2$.

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

(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 nine points and two constraining edges. Any constrained triangulation

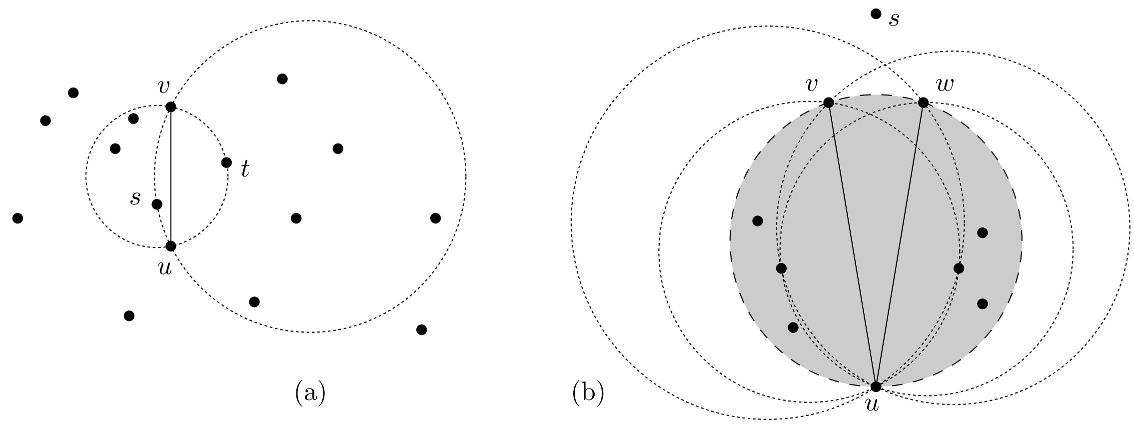

Figure 1: (a) Illustration of the first-points-hit (s and t). (b) Illustration of the proof.

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 four. 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, denoted $C(u, v, w)$, contains $2k-2$ points inside.

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 $\triangle uvw$ of it. The circle $C(u, v, w)$ can only contain points that are 'behind' edges of the constrained Delaunay triangulation, 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, it can have 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, and 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 of C_1 and C_2. 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 constrained Delaunay triangulation, part (i) of the lemma follows. □

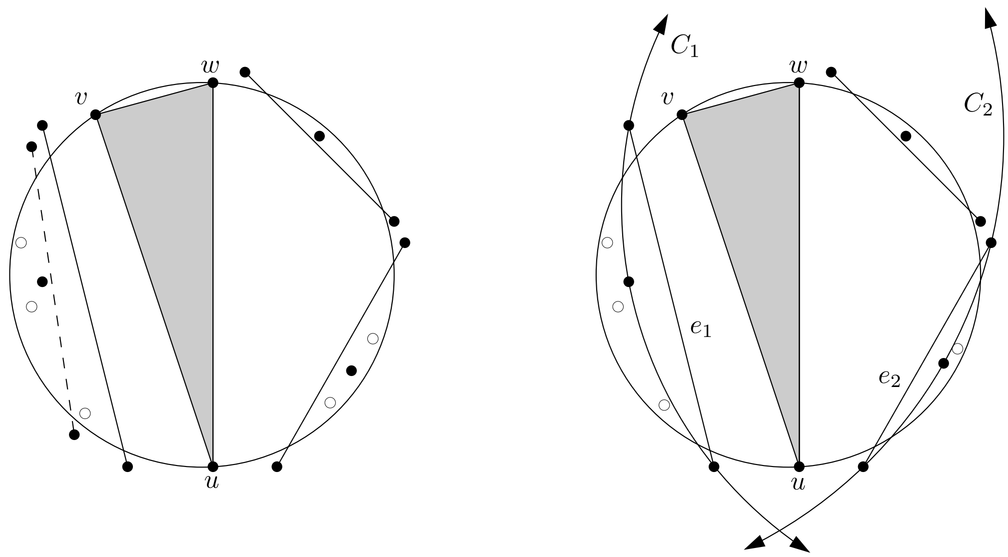

Figure 2: The order of a triangle in a constrained Delaunay triangulation.

The theorem implicitly shows that a set E of edges that are useful order-2 Delaunay edges can be completed to the lowest order Delaunay triangulation by computing the constrained Delaunay triangulation, because $2k-2=k$ for $k=2$. The case $k=3$ is the lowest order case where the theorem leaves a gap: we can complete to an order-4 Delaunay triangulation, but an order-3 Delaunay triangulation may exist too. We will show next that for this case, we can still compute the lowest order completion.

Lemma 1 *Let P be a set of points and let E be a set of useful order-3 Delaunay edges. If the constrained Delaunay triangulation has a triangle of order 4, then at least two of its edges are in E.*

Proof: We argue in the same way as in the proof of part (i) in the theorem above. Consider a triangle $\triangle uvw$, the circle $C(u,v,w)$, and the set E' of edges that intersect $C(u,v,w)$ twice and are closest to $\triangle uvw$. Only in the case that E' contains at least two edges we can have more than $k=3$ points inside $C(u,v,w)$. So let e_1, e_2, C_1 and C_2 be defined as in the proof of Theorem 2 as well. C_1 contains u, v, and w, unless one or two of these are endpoints of e_1. So circle C_1 can only contain up to $k-1=2$ points other than u,v,w if e_1 is an edge of $\triangle uvw$. The same reasoning applies to e_2 and C_2. Hence, we can only have four points inside $C(u,v,w)$ if at least two edges of $\triangle uvw$ are in $E' \subseteq E$. □

Due to this lemma, we have the following algorithm to compute the lowest order completion for $k=3$. Let P be a set of points and let E be a set of useful order-3 Delaunay edges. Compute the constrained Delaunay triangulation of P and E. Test the order of every triangle t. If it is order 4, then t must have two edges of E by the lemma. If the third edge is in E as well, then obviously four is the lowest order completion of P and E. If the third edge e is not in E, then let t' be the triangle on the other side of that edge. If we can flip the edge e, destroying t and t', to form two new triangles, then do it. Otherwise, four is the lowest order completion. If we could do a flip for all order-4 triangles, then determine the order of the resulting triangulation. If it is order 3, return this triangulation. Otherwise, return the constrained Delaunay triangulation (order 4 is optimal).

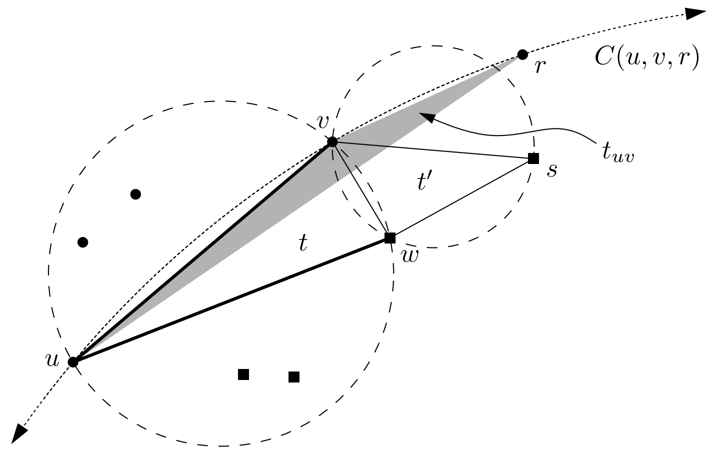

Figure 3: Illustration of the fact that if $\triangle uvs$ and $\triangle uws$ are not in the triangulation, it must have order at least four.

To prove that a lowest order completion is made, consider the case of an order-4 triangle $t = \triangle uvw$, let $\overline{vw}$ be the edge of t not in E, and let $t' = \triangle vws$, see Figure 3. Since the useful order of both $\overline{uv}$ and $\overline{uw}$ is at most three, and $C(u, v, w)$ contains four points, there must be exactly two points of P behind $\overline{uv}$ and exactly two points behind $\overline{uw}$, as in Figure 3.

Let T be any triangulation of P and E. If T has order 3, it cannot contain t, so consider the triangle t_{uv} with $\overline{uv}$ as an edge and which intersects t, and the triangle t_{uw} with $\overline{uw}$ as an edge and which intersects t. If the third vertex of both t_{uv} and t_{uw} is s, then an order-3 completion may be possible and we test it. So assume without loss of generality that $t_{uv} = \triangle uvr$ and $r \neq s$ (the grey triangle in Figure 3). Clearly $\overline{vs} \notin E$ because $\overline{ur}$ intersects this edge. But $\overline{vs}$ is an edge of the constrained Delaunay triangulation so $C(v, w, s)$ is void of points, except possibly behind edges of E that intersect the circle twice. Since $\overline{ur}$ does not intersect any edge of E but it intersects $\overline{vs}$, it follows that r must lie outside $C(v, w, s)$. But then $C(u, v, r)$, the circle for triangle $t_{uv} \in T$, must contain at least s, w, and the two points inside $C(u, v, w)$ that lie behind $\overline{uw}$. So $C(u, v, r)$ contains at least four points (squares in the figure), which contradicts the assumption that T has order 3.

We conclude with the following theorem. The time bound follows from Section 2.

Theorem 3 *A set P of n points and a set E of useful order-3 Delaunay edges with their endpoints in P can be completed to a lowest order Delaunay triangulation in $O(n \log n)$ time.*

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 criteria for realistic terrain modeling [10] 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. We can only do edges with useful orders up to $k = 3$. For higher orders we do not have any polynomial-time algorithm. Also, the constrained Delaunay triangulation gives a 2-approximation of the lowest order for completion, and it may be possible to improve upon this approximation factor.

References

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

[2] M. Bern, H. Edelsbrunner, D. Eppstein, S. Mitchell, and T. S. Tan. Edge insertion for optimal triangulations. *Discrete Comput. Geom.*, 10(1):47–65, 1993.

[3] L. P. Chew. Constrained Delaunay triangulations. *Algorithmica*, 4:97–108, 1989.

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

[5] O. Devillers, R. Estkowski, P.-M. Gandon, F. Hurtado, P. Ramos, and V. Sacristán. Minimal set of constraints for 2D constrained Delaunay reconstruction. *Int. J. of Comput. Geom. & Appl.*, 13:391–398, 2003.

[6] Nira Dyn, David Levin, and Samuel Rippa. Data dependent triangulations for piecewise linear interpolation. *IMA Journal of Numerical Analysis*, 10:137–154, 1990.

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

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

[9] 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.

[10] 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.