A Quadratic Time Algorithm for the MinMax Length Triangulation*
(extended abstract)

Herbert Edelsbrunner and Tiow Seng Tan

Department of Computer Science,
University of Illinois at Urbana-Champaign,
Urbana, Illinois 61801, USA.

Abstract

We show that a triangulation of a set of n points in the plane that minimizes the maximum edge length can be computed in time $O(n^2)$. The algorithm is reasonably easy to implement and is based on the theorem that there is a triangulation with minmax edge length that contains the relative neighborhood graph of the points as a subgraph. With minor modifications the algorithm works for arbitrary normed metrics.

1 Introduction

A *triangulation* of a (finite) point set S in $\Re^2$ is a maximally connected straight line plane graph whose vertices are the points of S. Maximality implies that with the exception of the unbounded face each face of the graph is a triangle. The number of different triangulations of S depends on $n = |S|$ as well as the relative location of the points. As implied by a result in [ACNS82], 10^{13n} is an upper bound on the number of triangulations of any set of n points in $\Re^2$. Furthermore, if S is in convex position then it admits $\frac{1}{n-1}\binom{2n-4}{n-2} \geq 2^{n-3}$ different triangulations. In order to choose an optimal triangulation, under some criterion, it is thus not feasible to exhaustively search the set of all triangulations.

Indeed, except for a handful of particular optimality criteria, the problem of finding an optimal triangulation for a given point set is hard, that is, no polynomial time algorithms are known. Among these exceptions are the maxmin angle criterion [Sibs78], the minmax angle criterion [EdTW90], the minmax smallest enclosing circle criterion [Raja91], and the minmax circumscribed circle criterion. The optimum under the first, third and fourth criterion is achieved by the Delaunay triangulation which can be constructed in time $O(n \log n)$ [Dela34, PrSh85, Edel87].

In this paper we study the complexity of minimizing the maximum edge length. A triangulation that minimizes the length of its longest edge is called a *minmax length triangulation*. It is related to the so-called *minimum length* (or *minimum weight*) *triangulation* that minimizes the sum of the edge lengths. The latter problem has been studied by Plaisted and Hong [PlHo87], Lingas [Ling87], and others. In spite of the lack of a proof that the problem is NP-hard, no polynomial time algorithm for constructing a minimum length triangulation is currently known. Even more annoying is the lack of a constant approximation scheme, that is, an algorithm that in polynomial time constructs a triangulation guaranteed to have total edge length at most some constant times the optimum. The currently best approximation scheme is described in [PlHo87] and guarantees a factor of $O(\log n)$.

In view of the apparent difficulty to compute minimum length triangulations, it is somewhat surprising that we are able to give a polynomial, in fact quadratic time algorithm for constructing a minmax length triangulation. It is the first polynomial time algorithm for this problem. There is evidence for the potential usefulness of such a triangulation (see [BrZl70, WiGS90]). Still, the authors of this paper consider the additional insight into optimum triangulations under edge length criteria as the main contribution here.

The reader might find it instructive to rule out seemingly promising approaches to computing minmax length triangulations before diving into the occasionally involved developments of the forthcoming sections. Note first that the Delaunay triangulation does not minimize the maximum edge length (see also Section 2). Second, the incremental greedy method, that repeatedly adds the shortest edge that does not intersect any previously added edge, also fails to minimize the maximum edge length. Third, let us take a brief look at the decremental greedy method that throws away edges in the order of decreasing length. It stops the deletion process if another deletion would render the set of edges so that it does not contain any triangulating subset (see Wismath [Wism80, page 81]). The trouble with this approach is that it is not clear how to efficiently decide whether the evolving edge set is still sufficient to triangulate the point set. Indeed, Lloyd [Lloy77] proves that the general version of this problem (decide whether a given edge set contains a triangulation) is NP-hard. Finally, the iterative methods that use the edge-flip [Laws77] or the more general edge-insertion operation [EdTW90] can get caught in local optima. The

*Research of the first author is supported by the National Science Foundation under grant CCR-8921421. The second author is on study leave from the National University of Singapore, Republic of Singapore.

CH3062-7/91/0000/0414$01.00 © 1991 IEEE

approach taken in this paper is entirely different from the above paradigms.

The organization of this paper is as follows. Section 2 reviews a few results on relative neighborhood graphs and other subgraphs of the Delaunay triangulation. Section 3 formulates the global algorithm; its straightforward implementation using dynamic programming takes time $O(n^3)$. The only intricate part of this algorithm is the proof of correctness provided in Section 4. Sections 5 and 6 present a specialized polygon triangulation algorithm that can be used to speed up the general algorithm to time $O(n^2)$. While Sections 2 through 6 assume that the Euclidean metric is used to measure length, Section 7 demonstrates that all results extend to general normed metrics. Indeed, the arguments in Sections 2 through 6 are axiomatically derived from a few basic lemmas in order to minimize the number of changes necessary to generalize the results. Finally, Section 8 briefly discusses the contributions of this paper and states some related open problems. The full details of this paper can be found in [EdTa91].

2 Preliminaries

The approach to constructing a minmax length triangulation taken in this paper first adds enough edges to decompose the plane into simple polygonal regions and then (optimally) triangulates these regions. Both Plaisted and Hong [PlHo87] and Lingas [Ling87] used this approach to compute approximations of the minimum length triangulation. In our case, the initial set of edges is provided by the (boundary of the) convex hull and the relative neighborhood graph of the point set S. The remainder of this section formally introduces these graphs, along with the Delaunay triangulation and the minimum spanning tree of S, and reviews some basic facts about their relationships. If x, y, z are three points in $\Re^2$ then xy denotes the relatively open line segment with endpoints x and y, $|xy|$ is its length, and xyz denotes the open triangle with vertices x, y, z.

The *Delaunay triangulation* of S, denoted by $dt(S)$, contains an edge ab, $a, b \in S$, if there is a circle through a and b so that all other points lie outside the circle. If the points are in general position then $dt(S)$ is indeed a triangulation.

As mentioned in the introduction, the Delaunay triangulation does not minimize the length of the longest edge. Take for example the points $a = (-2, 0)$, $b = (1, \sqrt{3})$, $c = (1, -\sqrt{3})$, $d = (2 - \epsilon, 0)$, with $0 < \epsilon < 1$. They form a convex quadrilateral $abdc$ and the Delaunay triangulation uses ad as the fifth edge. As ϵ approaches 0 the length of ad approaches $\frac{2}{\sqrt{3}}$ times the length of the longest edge in the alternative triangulation. Indeed, $\frac{2}{\sqrt{3}}$ is the worst possible ratio as can be shown using the result of [Raja91] that the Delaunay triangulation minimizes the radius of the maximum smallest enclosing circle, where the maximum is taken over all triangles. If the radius of this circle is 1 then the longest edge of the Delaunay triangulation has length at most 2. By the optimality result every minmax length triangulation has a smallest enclosing circle of radius at least 1 and therefore an edge of length at least $\sqrt{3}$ (see also [WiGS90]).

The *convex hull* of S is the smallest convex polygon that contains S. We define $ch(S)$ as the graph defined by the edges of this polygon. In the (degenerate) case where three or more collinear points lie on the boundary of this polygon we think of each such point as a vertex of the polygon. Thus, edges are taken only between adjacent collinear points. Each convex hull edge is an edge of *every* triangulation of S, and therefore also of every minmax length triangulation.

An edge ab belongs to the *relative neighborhood graph* of S, denoted by $rng(S)$, if

$$|ab| \leq \min_{x \in S - \{a,b\}} \max\{|xa|, |xb|\}.$$

This definition goes back to Toussaint [Tous80] who modified a similar definition by Lankford [Lank69] for use in pattern recognition. Alternatively, we can define the *lune* of ab as the set $\{x \in \Re^2 : \max\{|xa|, |xb|\} < |ab|\}$, and define $rng(S)$ as the set of edges ab whose lunes have empty intersection with S.

A *minimum spanning tree* of S, $mst(S)$, is a spanning tree of S that minimizes the total edge length; it also minimizes the maximum edge length.

All four graphs, $dt(S), ch(S), rng(S), mst(S)$, are plane and connected, and with the exception of $ch(S)$, they span S. Where convenient we will interpret these graphs as edge sets. Plainly, $ch(S) \subseteq dt(S)$, and as observed by Toussaint [Tous80], we also have $mst(S) \subseteq rng(S) \subseteq dt(S)$. Obviously, $ch(S) \subseteq mlt(S)$, for every minmax length triangulation $mlt(S)$, and we will show in Section 4 that there exists an $mlt(S)$ so that $rng(S) \subseteq mlt(S)$.

3 The Global Algorithm

As mentioned above there exists a minmax length triangulation $mlt(S)$ that contains all edges of $ch(S)$ and $rng(S)$. Because $ch(S) \cup rng(S)$ is a connected graph, it decomposes the convex hull of S into simple *polygonal regions*, which we define as open sets, that contain no points of S. It is thus natural to construct $mlt(S)$ by computing $ch(S) \cup rng(S)$ and then (optimally) triangulating each polygonal region.

Strictly speaking, however, the polygonal regions are not necessarily simple polygons in the usual sense of the term, although their interiors *are* simply connected. The difference is that the interior of the closure of a polygonal region is not necessarily the same as the region itself; it may contain edges of the region and it may be non-simply connected. The most effective way to deal computationally with this minor difficulty is to represent each edge by a pair of oppositely directed edges, and to represent the boundary of each region by the collection of directed edges for which the region lies on their left hand side. In effect, this means that we interpret each polygonal region as a genuine simple polygon, simply by pretending that its zero-width cracks are opened up a tiny amount. In most cases, this is a convenient interpretation and the notation will be adjusted accordingly. Only occasionally, the difference between a simple polygonal region and a simple polygon will be uncovered.

Let us now formally specify the algorithm and give a preliminary analysis.

Input. A set S of n points in $\Re^2$.
Output. A minmax length triangulation of S.
Process. 1. Construct $ch(S)$ and $rng(S)$.
2. Determine the polygonal regions defined by $ch(S) \cup rng(S)$.
3. Find a minmax length triangulation for each such polygonal region.

Step 1 can be carried out in time $O(n \log n)$ using results documented in [PrSh85] and [Supo83] (see also [JaKY90]). Using the standard quad-edge data structure of [GuSt85] for storing the plane graph $ch(S) \cup rng(S)$, step 2 can be accomplished in time $O(n)$. Finally, we can use dynamic programming to compute an optimal triangulation for each polygon in time cubic and storage quadratic in the number of its vertices (see [Klin80]). This adds up to time $O(n^3)$ and storage $O(n^2)$. The correctness of the algorithm will be established in the next section. Sections 5 and 6 will show how to speed up the algorithm to time $O(n^2)$ using a specialized polygon triangulation algorithm.

4 The Subgraph Theorem

The main result of this section is what we call the Subgraph Theorem which was announced earlier. We begin with two elementary geometry lemmas about distances between four points in convex and in non-convex position.

□-Lemma. For a convex quadrilateral $abcd$, we have $|ab| + |cd| < |ac| + |bd|$.

Proof. Let x be the intersection point of the two diagonals, ac and bd. Clearly, $|ab|+|cd| < (|ax|+|xb|)+(|cx|+|xd|) = |ac|+|bd|$. □

In words, the total length of the two diagonals of a convex quadrilateral always exceeds the total length of two opposite sides. This is true even if three of the four vertices are collinear. It implies that if one diagonal is no longer than one of the edges then the other diagonal is longer than the opposite edge.

Δ-Lemma. Let a, b, c, d be four distinct points so that the closure of the triangle abc contains d. Then $|ad| < \max\{|ab|, |ac|\}$.

Proof. If a, b, d, d are collinear the result is obvious. Otherwise, let d' be the intersection of the edge bc with the line passing through a and d, and note that $|ad| \leq |ad'|$. Of all points on bc only the endpoints can possibly maximize the distance to a. The assertion follows because if d' is an endpoint of bc then $d \neq d'$ and therefore ad is strictly shorter than ad'. □

Note that the length of the longest edge of any minimum spanning tree is no longer than the longest edge of any triangulation of S. This follows trivially from the fact that every triangulation contains a spanning tree. It is not very difficult to prove that the same is true for the relative neighborhood graph of S. First we need some notation. The circle with center x and radius ρ is denoted by (x, ρ), and the *bisector* of two points p and q is the set of points equidistant to both.

Length Lemma. Every triangulation of S contains an edge that is at least as long as the longest edge of $rng(S)$.

Proof. Let pq be the longest edge of $rng(S)$ and let $t(S)$ be an arbitrary triangulation of S. If $pq \in t(S)$ there is nothing to prove. Otherwise, pq intersects edges $r_1s_1, r_2s_2, \ldots, r_ks_k$ of $t(S)$, sorted from p to q, with all r_i on one side of the line through p and q and all s_i on the other. If pq is longer than all edges in $t(S)$ then r_1 and s_1 are both inside the circle $C_p = (p, |pq|)$, because pr_1 and ps_1 are both edges of $t(S)$. By the definition of $rng(S)$, r_1 and s_1 are thus outside or on the circle $C_q = (q, |pq|)$. Therefore, r_1 and s_1 lie in the half-plane of points closer to p than to q. Symmetrically, r_k and s_k lie inside C_q and outside or on C_p and therefore in the half-plane of points closer to q than to p. For each $1 \leq i \leq k-1$ we have either $r_i = r_{i+1}$ or $s_i = s_{i+1}$, which implies that there is an index j so that r_j and s_j do not lie on the same side of the bisector of pq. But then the □-Lemma implies that $|r_js_j| > |pq|$, because $|pq|$ is no longer than each of two opposite edges of the convex quadrilateral pr_jqs_j, a contradiction. □

We are now ready to state and prove the Subgraph Theorem. Its proof is similar to, although considerably more involved, than the proof of the Length Lemma. The basic idea is to assume an extreme counterexample and to contradict its existence by retriangulating parts of it.

Subgraph Theorem. Every finite point set S in $\Re^2$ has a minmax length triangulation $mlt(S)$ so that $rng(S) \subseteq mlt(S)$.

Proof. Let us assume there is a set S so that no minmax length triangulation contains $rng(S)$. We arbitrarily and independently index the points of S and the edges of $rng(S)$. Let $t(S)$ be a minmax length triangulation of S that satisfies the following extremal properties, where later properties are contingent upon earlier ones.

(i) $t(S)$ maximizes the smallest index of any edge in $rng(S)$ that is not in $t(S)$. Let pq be the edge with this index and let the index of p be smaller than that of q.

(ii) $t(S)$ minimizes the number of edges that intersect pq.

In case $t(S)$ is not unique yet we add two extremal properties that depend on points b and b' to be defined later. More precisely, b is an angle around a point, and so is b', but this will become clear shortly. The two points (or angles) are uniquely defined by pq and a given triangulation.

(iii) $t(S)$ minimizes the number of edges incident to b that intersect pq.

(iv) $t(S)$ minimizes the number of edges incident to b' that intersect pq.

It is conceivable that $t(S)$ is still not unique, but it will be sufficient to assume that $t(S)$ is any one of the remaining triangulations.

Consider the edge pq of $rng(S)$; it intersects the triangles $t_1, t_2, \ldots, t_k$ of $t(S)$, sorted from p to q (see Figure 4.1 left). By deleting the edges that intersect pq we create a polygonal region. As mentioned in Section 3 we interpret each edge in its boundary as a pair of edges with opposite direction, and to trace the boundary of the region we traverse all directed

edges that have the region on their left side. This allows us to think of the polygonal region as a simple polygon. Any two consecutive (directed) edges define an *angle* (see Figure 4.1 middle). Note that a vertex can correspond to many angles, although the common situation is that it corresponds only to one. We will therefore sometimes ignore the difference between vertices and corresponding angles. Points p and q correspond to only one angle each. An angle is *convex* if the two defining edges form a left-turn. Call the sequence of edges from p to q the *lower chain* and the sequence from q to p the *upper chain*. Each chain contains at least one convex angle different from p and q.

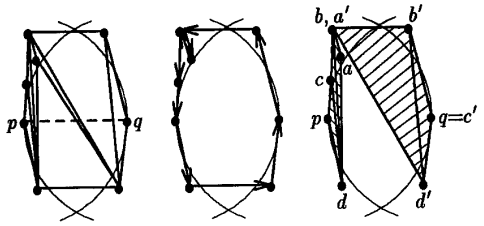

Figure 4.1: To the left we see the triangles of $t(S)$ that intersect pq. If we remove the edges intersecting pq we get a polygon whose boundary is oriented in a counterclockwise order. The prefix P and the suffix Q defined for this configuration are illustrated to the right. Although b and a' are the same point, they refer to different angles of this point.

A *prefix* is an initial subsequence of $t_1, t_2, \ldots, t_k$, and a *suffix* is a terminal subsequence of $t_1, t_2, \ldots, t_k$. We say that a prefix (suffix) *covers* an angle of the polygon if it contains all triangles incident to this angle. Let i be minimal so that the prefix $P = t_1, t_2, \ldots, t_i$ covers a convex angle other than p, and let j be maximal so that the suffix $Q = t_j, t_{j+1}, \ldots, t_k$ covers a convex angle other than q. P and Q consist of at least two triangles each. We let b be the convex angle (vertex) covered by P — it is incident to both t_i and t_{i-1} — and d be the other vertex common to t_i and t_{i-1}. Furthermore, c is the third vertex of t_{i-1} and a is the third vertex of t_i (see Figure 4.1 right). Symmetrically, define vertices b', d', c', a' of Q. We say that P (Q) is *type* 1 if the last (first) two triangles of P (Q) are the only ones incident to b (b'), and it is *type* 2, otherwise (see Figure 4.2). If P is type 1 then a, b, c belong to the same chain and d belongs to the other chain (this includes the case that $c = p$), and

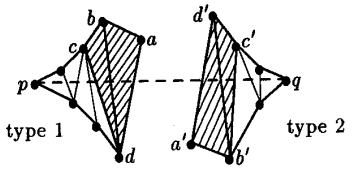

Figure 4.2: The prefix P with vertices a, b, c, d and the suffix Q with vertices a', b', c', d' are defined depending on pq. P is type 1 and Q is type 2. For illustration purposes the constraint that all vertices must lie outside the lune of pq has been ignored.

if P is type 2 then a, b belong to one chain and c, d to the other.

Claim 1. $P = t_1, t_2, \ldots, t_i$ **and** $Q = t_j, t_{j+1}, \ldots, t_k$ **share at most two triangles, that is,** $i - 1 \leq j$.

Proof (of Claim 1). We show that the suffix $R = t_{i-1}, t_i, \ldots, t_k$ covers at least one convex angle other than q, so Q cannot be bigger than R. If P is type 1 then R covers b, which is convex. Otherwise, R covers all angles between d and q, d included. Since all angles between p and d, p and d excluded, are non-convex, at least one angle between d and q must be convex, and this angle is covered by R. This completes the proof of Claim 1.

In order to get a contradiction to the choice of $t(S)$ we attempt a retriangulation of the polygonal region defined by pq, using P and Q. Of course, the maximum edge length must not increase. We show below that either bd, or $b'd'$, or both can be switched, which will lead to a contradiction except in a particular case where P and Q share one triangle. The analysis of this case will conclude the proof.

Call bd ($b'd'$) *switchable* if ac ($a'c'$) is no longer than the longest edge of $t(S)$. We are able to prove strong locality constraints for a and d (a' and d') if bd ($b'd'$) is not switchable. Define

$$A = \{x \in \Re^2 : |xp| > |pq| \text{ and } |xp| > |xq|\} \text{ and}$$

$$D = \{x \in \Re^2 : |xp| \geq |pq| \text{ and } |xq| < |pq|\},$$

with the understanding that A and a belong to one half-plane defined by the line passing through p and q, and D and d belong to the other (see Figure 4.3).

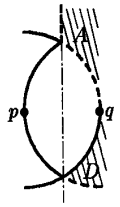

Figure 4.3: The regions A and D as defined for the case when a is on the upper chain.

Claim 2. **If** bd **is not switchable then** $a \in A$ **and** $d \in D$.

Proof (of Claim 2). Since bd is not switchable ac must be longer than the other five edges defined by a, b, c, d, and, by the Length Lemma, it must be longer than pq. We first show that $|ac| \leq |ap|$ and then derive the four inequalities needed to establish the claim.

(1) $|ac| \leq |ap|$. We can assume that $c \neq p$. Note that c is contained in the closure of triangle bdp. Since the line passing through b and d separates a from p, the closures of the two triangles abp and adp cover bdp completely, and therefore one of them contains c. If c lies in the closure of abp the claim follows from $|ab| < |ac|$ and the Δ-Lemma for abp, and if c lies in adp if follows from $|ad| < |ac|$ and the Δ-Lemma for adp.

(2) $|ap| > |pq|$. From the Length Lemma we get $|pq| < |ac|$ and from (1) we get $|ac| \leq |ap|$.

(3) $|dq| < |pq|$. Assume $|dq| \geq |pq|$. The $\square$-Lemma for $paqd$ implies $|ad| > |ap|$ and thus $|ad| > |ac|$ because of (1), a contradiction.

(4) $|dp| \geq |pq|$. This is immediate from (3) because pq is an edge of $rng(S)$.

(5) $|ap| > |aq|$. Assume $|ap| \leq |aq|$ and recall $|dp| \geq |pq|$ from (4). By the $\square$-Lemma for $paqd$ we get $|ad| > |aq|$, which implies $|ad| > |ap|$ by assumption, and $|ad| > |ac|$ by (1), a contradiction.

The proof of Claim 2 is now complete because (2) and (5) are equivalent to $a \in A$ and (3) and (4) are equivalent to $d \in D$.

Symmetrically, we define regions A' and D' which is where a' and d' must lie if $b'd'$ is not switchable. Using Claims 1 and 2 we can now show that there is always an edge that can be switched.

Claim 3. It is not possible that both bd and $b'd'$ are non-switchable.

Proof (of Claim 3). If bd and $b'd'$ are both non-switchable, then ad lies on q's side of the bisector of pq and $a'd'$ lies on p's side, by Claim 2. Because of Claim 1 and because ad is the last edge of P and $a'd'$ is the first edge of Q we have $\{a, d, a', d'\} = \{a, b, c, d\} = \{a', b', c', d'\}$. Furthermore, the fact that bd and $b'd'$ are both edges of $t(S)$ implies that they are the same and thus $b = d'$, $d = b'$, $a = c'$, $c = a'$ (see Figure 4.4). It follows that the polygonal region has the shape of a diamond with p, b, q, d as the only convex angles. This contradicts the locality constraints for a, b, c, d stated in Claim 2. In particular, the chain from p to $d \in D$ (as indicated by the dotted chain in Figure 4.4) is concave or straight and therefore enclosed by the circle $(q, |pq|)$. It follows that this chain is disjoint from A', which is where $c = a'$, the predecessor of d in this chain, is supposed to lie. The proof of Claim 3 is thus complete.

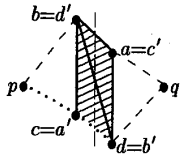

Figure 4.4: If bd and $b'd'$ are both non-switchable then b and d are the only convex angles besides p and q.

Let us now see how Claim 3 can be used to contradict the existence of $t(S)$ and thus of a counterexample to the assertion of the Subgraph Theorem. If bd is switchable and P is type 1 then the number of edges that intersect pq decreases when bd is switched. This contradicts property (ii). Thus, P must be type 2 if bd is switchable, and, similarly, Q must be type 2 if $b'd'$ is switchable. When we switch bd the degree of b decreases, which contradicts property (iii). Thus, it must be that bd is not switchable and $b'd'$ is. But switching $b'd'$ decreases the degree of b', which would contradict property (iv), unless the degree of b increases at the same time. Remember that (iv) is contingent upon (iii), so if (iii) is not satisfied any more then we cannot draw any

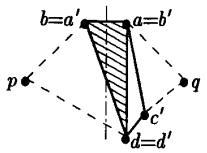

Figure 4.5: In the final configuration bd is non-switchable, so $a \in A$ and $d \in D$, and $b'd'$ is switchable, so Q is type 2. Furthermore, switching $b'd'$ to $a'c'$ increases the degree of b, so $a' = b$ and therefore P and Q overlap in exactly one triangle. The figure ignores that by rights all points should lie outside the lune of pq.

conclusion. Thus, the configuration left for analysis is as shown in Figure 4.5.

To reach the final contradiction, we switch $b'd'$ and redefine Q based on the new configuration. Since all angles from (the old) d' to q are non-convex, the new points b' and a' are the same as before, and the new d' is the old c'. Thus, we can again switch $b'd'$, and so on, until Q is type 1 or $c' = q$ at which point the next switch decreases the number of edges intersecting pq. This finally contradicts property (ii). □

Remark. A natural extension of minimizing the length of the longest edge in a triangulation is to also minimize the length of the second longest edge, and so on. Let $mvt(S)$ be a triangulation that minimizes the entire vector of edge-lengths in this fashion. If the points of S are in general position then $mvt(S)$ is unique. Curiously, it is not always true that (there is an) $mvt(S)$ (that) contains $rng(S)$ as a subgraph. The smallest example that illustrates this observation consists of four points a, b, c, d so that c and d lie fairly close to b, ab and cd intersect, and c and d both lie outside the circle $(a, |ab|)$.

5 Triangulating rng-Polygons

The goal of this section and the next is to improve the cubic time algorithm of Section 3 to quadratic time. This is done using a specialized polygon triangulation algorithm. The main part of the algorithm, and the structural properties of minmax length triangulations that guarantee its correctness, are developed in this section.

Recall that the first two steps of the algorithm in Section 3 decompose the convex hull of S into polygonal regions by drawing all edges of $ch(S)$ and $rng(S)$; these steps remain unaltered. Each region is represented by a cyclic chain of directed edges that trace its boundary in a counterclockwise order around the region. Because $rng(S)$ is a connected graph which spans S, any polygonal region is bounded by at most one edge not in $rng(S)$; this edge is in $ch(S) - rng(S)$. We call a polygonal region a *complete rng-polygon* if all its edges belong to $rng(S)$, and an *incomplete rng-polygon*, otherwise.

Obviously, rng-polygons are not as general as arbitrary polygonal regions because for each edge ab, except possibly for one, the lune of ab, $\lambda_{ab} = \{x \in \Re^2 : \max\{|ax|, |bx|\} <$

$|ab|\}$, is free of points of S. We call pq a *diagonal* of a polygonal region if it lies in the region entirely. For each diagonal pq of an rng-polygon it must be that λ_{pq} contains at least one point of S. We further distinguish between the cases where λ_{pq} contains points of S on both sides of pq and where it does not.

For a directed edge $\vec{pq}$ let $h_{\vec{pq}}$ be the set of points to the left of or on the directed line that passes through p and q in this order. Define the *half-lune* of $\vec{pq}$ as

$$\eta_{\vec{pq}} = \lambda_{pq} \cap h_{\vec{pq}}.$$

By definition, $\lambda_{pq} = \eta_{\vec{pq}} \cup \eta_{\vec{qp}}$, and we have $pq \in rng(S)$ iff $\eta_{\vec{pq}} \cap S = \eta_{\vec{qp}} \cap S = \emptyset$. We call pq a *2-edge* if both half-lunes contain points of S, and we call it a *1-edge* if only one half-lune contains points of S. For a 1-edge pq, we say the side where the half-lune contains points of S is *beyond* pq, and the other side is *beneath* pq. Note for example that if pq is a 1-edge bounding an incomplete rng-polygon R then $pq \in ch(S)$ and therefore R is beyond pq. We will see later that 1-edges are useful in triangulating rng-polygons.

The first lemma of this section shows that when we triangulate an rng-polygon R, whether complete or incomplete, we can ignore all points outside R. More specifically, it shows that the type of any diagonal or edge of R remains unchanged when we remove all points of S that are not vertices of R.

Reduction Lemma. Let pq be a diagonal or edge of an rng-polygon R. If $\eta_{\vec{pq}}$ contains points of S then it also contains vertices of R.

Proof. Assuming $\eta_{\vec{pq}}$ contains points of S but no vertices of R, it must intersect edges of R without containing their endpoints. Let yy' be the edge closest to p and q, and let x be a point in $\eta_{\vec{pq}} \cap S$. Since x is not a vertex of R it must lie on the other side of yy', as seen from p and q. So $yy' \in rng(S) - ch(S)$, and therefore $\max\{|xy|, |xy'|\} \geq |yy'|$. Assume without loss of generality that $|xy| \geq |yy'|$. If y' lies outside or on the circle $(p, |pq|)$ we consider the convex quadrilateral $pyxy'$. Otherwise, y' lies outside or on $(q, |pq|)$ in which case we consider the convex quadrilateral $qyxy'$. But now we have $|xy| \geq |yy'|$ and either $|py'| > |px|$ or $|qy'| > |qx|$, a contradiction to the $\square$-Lemma in both cases. □

Using the Reduction Lemma we now address vertices visible from both endpoints of an edge. We need some notation. Two points x, y inside or on the boundary of a polygonal region are *visible* from each other if xy is contained in the region. The *distance* of a point x to an edge pq is defined as the infimum, over all points $z \in pq$, of $|xz|$. If $|pq| > \max\{|px|, |qx|\}$ then this distance is referred to as the *height* of the triangle pqx.

Visibility Lemma. Let pq be a diagonal or edge of an rng-polygon R, and let x be a vertex of R that lies in $\eta_{\vec{pq}}$ and minimizes the distance from pq. Then x is visible from p and also from q.

Proof. Consider the triangle pqx, let $x' \in pq$ be the point with minimum distance from x, and assume without loss of generality that x is not visible from q. Let yy' be an edge of R that intersects qx. The proof of the Reduction Lemma implies that at least one endpoint of yy' lies in $\eta_{\vec{pq}}$, say $y \in \eta_{\vec{pq}}$. In addition, y and y' lie outside the triangle pqx because x is closest to pq (see Figure 5.1). Hence, yy' intersects xp, xq and all edges xz with $z \in pq$. Thus, $xyx'y'$ is a convex quadrilateral, and because of $|yx'| \geq |xx'|$ by the choice of x, we have $|yy'| > |y'x|$ from the $\square$-Lemma. By symmetry, if y' lies in $\eta_{\vec{pq}}$ we have $|yy'| > |xy|$, which implies $yy' \notin rng(S)$. This is a contradiction because $yy' \notin ch(S)$. Thus, y' must lie outside $\eta_{\vec{pq}}$. If y' lies outside or on the circle $(p, |pq|)$ then $|py'| > |px|$ and therefore $|xy| < |yy'|$ by the $\square$-Lemma for $py'xy$. Symmetrically we get $|xy| < |yy'|$ from the $\square$-Lemma for $qy'xy$ if y' lies outside or on the circle $(q, |pq|)$. Together with $|xy'| < |yy'|$ this contradicts $yy' \in rng(S)$. □

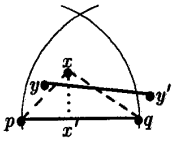

Figure 5.1: The quadrilateral $xyx'y'$ is convex because $x' \in pq$ and $y, y' \notin pqx$.

We need one more elementary lemma.

Containment Lemma. If $x \in \eta_{\vec{pq}}$ then $\eta_{\vec{xp}} \subseteq \lambda_{pq}$.

Proof. Take a point $z \in \eta_{\vec{xp}}$ and consider the four points p, q, x, z. If $z \in pq$ there is nothing to prove. Otherwise, $pzqx$ or $pqzx$ is a convex quadrilateral (possibly with three of the four vertices collinear) or $z \in pqx$. In each case $|qz| < |pq|$ can be shown using the $\square$- or the Δ-Lemma. This implies $z \in \lambda_{pq}$. □

The following lemma is of fundamental importance to the quadratic time triangulation algorithm.

1-Edge Lemma. Let pq be a 1-edge of an rng-polygon R, and let x be a vertex of R that lies in $\eta_{\vec{pq}}$ and minimizes the distance from pq. Then px is either an edge of R or a 1-edge with pqx beneath px, and the same is true for qx.

Proof. We have $\eta_{\vec{xp}} \subseteq \lambda_{pq}$ by the Containment Lemma. The part of $\eta_{\vec{xp}}$ in $\eta_{\vec{qp}}$ contains no point of S because $\eta_{\vec{qp}} \cap S = \emptyset$ by assumption. For a different reason also the part of $\eta_{\vec{xp}}$ in $\eta_{\vec{pq}}$ contains no point of S. This is because a point $y \in \eta_{\vec{xp}} \cap \eta_{\vec{pq}}$ would be closer to pq than x is, as can be shown using the $\square$-Lemma for $px'yx$ (see Figure 5.2). So px is an edge of R if $\eta_{\vec{px}}$ contains no point of S either, and it is a 1-edge with triangle pqx on its beneath side, otherwise. The argument for qx is symmetric. □

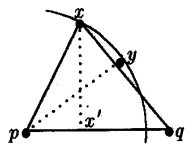

Figure 5.2: Vertex x is visible from p and from q, so pqx is empty. It follows that if $y \in \eta_{\vec{xp}} \cap \eta_{\vec{pq}}$ then $pqyx$ is a convex quadrilateral.

5.1 Incomplete *rng*-Polygons

The above lemmas are sufficient for efficiently triangulating an incomplete *rng*-polygon. As defined earlier, all edges of an incomplete *rng*-polygon R are *rng*-edges, except for one 1-edge, $pq \in ch(S) - rng(S)$, which has R on its beyond side. The algorithm below can triangulate more general incomplete *rng*-polygons, that is, it is not necessary that $pq \in ch(S)$, but it must be that pq is a 1-edge and R lies beyond pq.

Input. An incomplete *rng*-polygon R that lies beyond its 1-edge pq.
Output. A minmax length triangulation of R.
Process.
1. Find a vertex x in λ_{pq} that minimizes the distance from pq.
2. Draw edges px and qx. This decomposes R into the triangle pqx, and two possibly empty incomplete *rng*-polygons R_1 and R_2.
3. Recursively triangulate R_1 and R_2.

The correctness of this algorithm follows from the 1-Edge Lemma. Indeed, it implies that if R_1 is non-empty then it lies beyond px, which is the only 1-edge of R_1. Similarly, R_2 lies beyond its 1-edge qx, provided R_2 is non-empty. Thus, the input invariant is maintained all the way through the recursion. This implies that the algorithm successfully triangulates. By the choice of point x, the edges px and qx are both shorter than pq. It follows that the diagonals are monotonely decreasing in length, down a single branch of the recursion, and therefore all diagonals constructed by the algorithm are shorter than pq. A straightforward implementation of the algorithm takes time quadratic in the number of vertices of R.

Remark. Instead of choosing a vertex x that minimizes the distance to pq, step 1 of the algorithm could also choose other vertices as long as they are visible from p and q and lie in their lune. An interesting choice among these vertices is the vertex y that minimizes $\max\{|yp|, |yq|\}$. As long as y is unique, which is the non-degenerate case, this choice leads to a triangulation of the polygon R that lexicographically minimizes the sorted vector of edge lengths. Another possible choice is the vertex z that minimizes $|zp| + |zq|$. This vertex is automatically visible from p and from q and might be useful in actual implementations because it is often considerably less expensive to compute the distance between two points than between a point and a line segment.

5.2 On Polygon Retriangulation

This subsection presents a technical lemma on retriangulating a polygonal region. It will find application in Sections 5.3 and 6, and is also of independent interest. In order to conveniently distinguish between boundary and non-boundary edges of a triangulation, we call a non-boundary edge a *diagonal*. Let X be a polygonal region, $t(X)$ a triangulation of X, and xx' a diagonal of X that is not in $t(X)$. We say that xx' *generates* $t(X)$ if it intersects every diagonal of $t(X)$. We give an algorithmic description of a particular triangulation of X, called the *fan-out triangulation* $f_x(X)$ with *(fan-out) center* x. The triangulation is illustrated in Figure 5.3.

1. Connect x to all vertices of X that are visible from x. Call these vertices and also the two vertices connected to x by edges of X *neighbors* of x.
2. Two neighbors of x are said to be *adjacent* if they are consecutive in the angular order around x. Connect any two adjacent neighbors u, v of x, unless uv is an edge of X.
3. Every edge uv created in step 2 decomposes X into two parts, and the part that does not contain x is called the *pocket* X_{uv} of uv. Assume that u is the endpoint of uv so that the other incident edge of the pocket, uw, is partially visible from x. Recursively construct the fan-out triangulation of X_{uv} with center v.

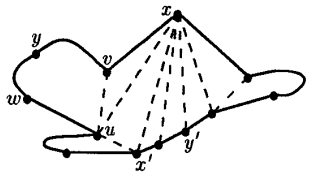

Figure 5.3: The polygonal region X is triangulated by fanning out from x, connecting adjacent neighbors of x, and recursing in the thus created pockets.

Fan-Out Lemma. Let X be a polygonal region, with δ_1 the length of its longest edge, let $t(X)$ be a triangulation of X, with δ_2 the length of its longest diagonal, let xx' be a generator of $t(X)$, and let δ_3 exceed the maximum distance of x from any vertex of X. Then $|ab| < \max\{\delta_1, \delta_2, \delta_3\}$ for every diagonal ab of $f_x(X)$.

The proof can be found in [EdTa91].

5.3 Complete *rng*-Polygons

It will be convenient to assume that no two diagonals and edges of the *rng*-polygon R are equally long. With this assumption we can show that every triangulation of R, and therefore also every minmax length triangulation, contains a 2-edge. To see this take the longest edge pq of a triangulation. It is not an edge of R because the third vertex of the incident triangle lies in its lune λ_{pq}. It is therefore a diagonal with incident triangles pqr and pqs, and we have $r, s \in \lambda_{pq}$ by maximality of pq. Since r and s lie on different sides of pq it follows that pq is a 2-edge.

We prove below that there is a minmax length triangulation $mlt(R)$ of R that contains only one 2-edge pq. By the argument above pq is the longest edge of $mlt(R)$. We call pq *expandable* if there are vertices r and s in λ_{pq}, on different sides of pq and both visible from p and q, so that $E = \{pr, qr, ps, qs\}$ is a set of *rng*- and 1-edges and the quadrilateral $prqs$ lies beneath the 1-edges in E. It should be clear that once we draw an expandable 2-edge we can complete the triangulation using the algorithm for incomplete *rng*-polygons (Section 5.1). The resulting triangulation uses no 2-edge other than pq which is thus the longest edge of the triangulation.

We first present the algorithm and then prove its correctness by showing that every complete rng-polygon R has a minmax length triangulation that contains an expandable 2-edge. This, however, assumes that no two diagonals or edges of R have equal length. If this non-degeneracy constraint is not satisfied it is necessary to run the algorithm with a simulation of non-degeneracy, see [EdMü90]. The side-effects of this simulation and how they can be undone are discussed in the full paper.

Input. A complete rng-polygon R.
Output. A minmax length triangulation of R.
Process.
1. Find the shortest expandable 2-edge pq, together with corresponding rng- and 1-edges pr, qr, ps, qs.
2. Triangulate the incomplete rng-polygons defined by pr, qr, ps, qs.

As mentioned in Section 5.1, step 2 takes time that is only quadratic in the number of vertices of R. In Section 6 we will see how step 1 can be implemented so it runs in quadratic time too. We now formulate and prove the lemma that implies the correctness of the algorithm.

2-Edge Lemma. *Let R be a complete rng-polygon with no two diagonals or edges of the same length. Then there exists a minmax length triangulation $mlt(R)$ of R that contains an expandable 2-edge.*

Proof. We assume there is no minmax length triangulation of R that contains an expandable 2-edge. A contradiction to this assumption will be derived using an extreme minmax length triangulation $t(R)$ defined as follows. Let pq be the longest edge of $t(R)$ and let pqr and pqs be the incident triangles. By the non-degeneracy assumption, pq is the longest edge of *every* minmax length triangulation of R. Choose $t(R)$ so that the sum of heights of pqr and pqs (that is, the distance of r from pq plus the distance of s from pq) is a minimum. We prove below that pq is expandable and that r and s are witnesses thereof, that is, the quadrilateral $prqs$ lies beneath every 1-edge in $E = \{pr, qr, ps, qs\}$.

Case 1. Assume that $prqs$ lies beyond at least one 1-edge in E, say beyond pr. Then we can retriangulate R on this side of pr using the algorithm for incomplete rng-polygons. Among others, this algorithm removes edge pq, and all new edges are shorter than pr, which itself is shorter than pq. This contradicts the assumption that $t(R)$ is a minmax length triangulation.

Case 2. Assume that one of the edges of E, say pr, is a 2-edge, and assume without loss of generality that $r \in \eta_{\overline{pq}}$. Thus, there is a non-empty set of vertices z of R contained in the half-lune $\eta_{\overline{rp}}$. By the Containment Lemma these vertices z lie in λ_{pq}, and by the Visibility Lemma a non-empty subset S' of the z are visible from both p and r.

If a vertex z is in S' then either $pz \cap rq \neq \emptyset$ or $rz \cap pq \neq \emptyset$, see Figure 5.4. Let S'_p be the subset of vertices z of the first kind, and let S'_r be the subset of vertices of the second kind. If $S'_p \neq \emptyset$ choose $x \in S'_p$ so that the number of edges of $t(R)$ that intersect px is a minimum. Next, remove all edges from $t(R)$ that intersect px and denote by X the polygonal region thus generated. If, on the other hand, $S'_p = \emptyset$, then choose $x \in S'_r \neq \emptyset$ so that the number of edges in $t(R)$ that intersect rx is a minimum, again remove all edges from

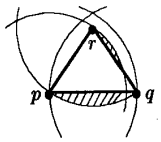

Figure 5.4: The points z lie in the interior of $\eta_{\overline{rp}} - pqr$, which consists of one or two connected components depending on whether or not the angle at r in the triangle pqr is non-acute.

$t(R)$ that intersect rx, and denote the resulting polygonal region by X. For convenient reference we set $x' = p$ in the first case and $x' = r$ in the second. In either case, we construct a retriangulation $f_x(X)$ of X by fanning out from x, as described in Section 5.2.

We show below that the new triangulation of R has properties that contradict the assumptions of case 2. Most importantly, the Fan-Out Lemma of Section 5.2, together with a few claims which we are about to prove, imply that the edges of $f_x(X)$ do not exceed pq in length.

Claim 2.1. *Except for x, all vertices of X lie outside the half-lune $\eta_{\overline{rp}}$.*

Proof (of Claim 2.1). Let $y_1y_2, y_3y_4, \ldots, y_{m-1}y_m$ be the edges, sorted from x' to x, that are removed from $t(R)$ when X is constructed. Suppose the claim is not true. Then there is a smallest index $j \leq m-1$ with $y_{j+1} \in \eta_{\overline{rp}}$. Consider the polygonal region X_j of $t(R)$ that is created by removing the edges $y_1y_2, y_3y_4, \ldots, y_{j-1}y_j$ from $t(R)$. Since y_{j+1} is the only vertex of X_j that lies in $\eta_{\overline{rp}}$ it is visible from p *and* from r, inside X_j. But this means that $y_{j+1}x'$ intersects fewer edges of $t(R)$ than xx'. This contradicts the choice of x and completes the proof of Claim 2.1.

Claim 2.2. *For each vertex y of X we have $|xy| < |pq|$.*

Proof (of Claim 2.2). Clearly, both px and rx are shorter than pq. So let y be any vertex different from p, r, x, and let yy' be an edge of $t(R)$ that intersects $x'x$. Because of Claim 2.1, x is visible within X from p and also from r, so $pyxy'$ and $ryxy'$ are convex quadrilaterals. Since y' lies outside $\eta_{\overline{rp}}$ it cannot lie inside both of the circles $(p, |pr|)$ and $(r, |pr|)$. If y' lies inside $(r, |pr|)$ then $|py'| > |px|$ which implies $|yy'| > |xy|$ by the $\square$-Lemma for $pyxy'$. Otherwise, we have $|ry'| > |rx|$ which implies $|yy'| > |xy|$ by the $\square$-Lemma for $ryxy'$. This concludes the proof of Claim 2.2 because yy' is an edge of $t(R)$ and is therefore no longer than pq.

Claim 2.2 and the Fan-Out Lemma imply that all diagonals of $f_x(X)$ are shorter than pq. In the case where $pq \cap rx \neq \emptyset$ we now have a contradiction, because the retriangulating process of X eliminates pq and all edges of the resulting new triangulation of R are shorter than pq. In the case where $rq \cap px \neq \emptyset$ the new triangulation still includes pq. We show below that the height of the new triangle incident to pq is smaller than the height of pqr and thus arrive at a contradiction.

So assume $rq \cap px \neq \emptyset$; in this case pq is an edge of the boundary of X and p is visible from x. If q is also visible from x then the new triangle incident to pq is pqx with height $|xx'|$, where $x' \in pq$ minimizes the distance to x.

Analogously define $r' \in pq$ that minimizes the distance to r. Since $|pr| > |px|$ we have $|rr'| > |xr'|$ by the $\square$-Lemma for $prxr'$. Together with $|xr'| \geq |xx'|$ this implies $|rr'| > |xx'|$. If q is not visible from x then pq belongs to the pocket X_{uv} defined by a cut-off edge uv. We have $u = p$, $w = q$, and the center v of X_{uv} lies inside pqx. So again, either pqv is a triangle, and its height is less than that of pqx and therefore that of pqr, or q is not visible from v, in which case the argument can be repeated. Eventually, we arrive at a triangle incident to pq whose height is less than that of pqr. $\square$

6 Finding the Shortest Expandable 2-Edge

This section shows how the first step of the algorithm for triangulating a complete *rng*-polygon R can be made to run in time $O(n^2)$, where n is the number of vertices of R. As in Section 5.3, we assume that no two diagonals or edges of R are equally long; so the shortest expandable 2-edge is unique. For convenience we also assume that no three vertices of R are collinear.

Input. A complete *rng*-polygon R.
Output. The shortest expandable 2-edge of R.
Process.
1. Determine the type of each diagonal pq of R.
2. For each 2-edge pq find vertices p', p'', q', q'' that minimize the counterclockwise angles $\angle p'pq, \angle qpp'', \angle q'qp, \angle pqq''$, contingent upon pp', pp'', qq', qq'' being *rng*-edges or 1-edges with pq on their beneath sides (see Fig. 6.1).
3. Return the shortest 2-edge pq for which pp', qq', pp'', qq'' are such that $p' = q''$ or $pp' \cap qq'' \neq \emptyset$, and $p'' = q'$ or $pp'' \cap qq' \neq \emptyset$.

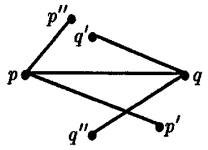

Figure 6.1: By the choice of p' the counterclockwise angle $\angle p'pq$ contains no 1-edge with pq on its beneath side. Symmetric statements hold for p'', q', and q''.

Below we give the algorithmic details of the above steps. **Step 1**, *classifying diagonals*. For each vertex p of R, we compute all incident diagonals pq and their angular order around p. Furthermore, we determine whether or not the half-lune $\eta_{p\bar{q}}$ contains any vertex of R. Recall that by the Visibility Lemma $\eta_{p\bar{q}}$ contains a vertex visible from p if it contains a vertex of R at all. We can thus base the decision whether or not $\eta_{p\bar{q}}$ is empty of vertices solely on the vertices visible from p. As defined earlier, pq is a 2-edge if both half-lunes of pq contain vertices of R. Otherwise, pq is a 1-edge and its beyond side is where the half-lune contains vertices of R. We now show that the computation for p can be done in time $O(n)$. It follows that $O(n^2)$ time suffices for step 1.

Computing the sorted sequence of diagonals $pp_1, pp_2, \ldots, pp_m$ incident to p is a standard operation for simple polygons and can be done in time $O(n)$, see e.g. [ElAv81]. Let pp_0 and pp_{m+1} be the two edges of R incident to p and assume that $p_0, p_1, p_2, \ldots, p_m, p_{m+1}$ is in a counterclockwise order around p. To determine whether there is a vertex of R in the half-lune $\eta_{p\bar{p}_i}$ for $1 \leq i \leq m$, we scan the list $p_0, p_1, \ldots, p_{m+1}$ once, from smallest index to largest. During the scan we maintain a stack of diagonals pp_l whose half-lunes $\eta_{p\bar{p}_l}$ are not yet found to contain any vertex of R. Before pushing pp_i onto the stack, we remove all diagonals pp_l whose half-lunes contain p_i. Using a straightforward extension of the Containment Lemma we can show that the order of processing implies that the edges whose half-lunes contain p_i lie on top of the ones whose half-lunes do not contain p_i. Thus, the former can be removed simply by repeatedly popping the topmost diagonal. When the scan is complete, the stack contains exactly all diagonals pp_l whose half-lunes contain no vertex of R. Since a diagonal can be pushed and popped only once each, the entire process takes constant time per diagonal.

Step 2, *finding rng- and 1-edges*. For each vertex p, we scan $pp_1, pp_2, \ldots, pp_m$ in this order. In the process we keep track of the most recent *rng*-edge or 1-edge $p\bar{p}$ whose beneath side is in the direction of the scan. Initially, $p\bar{p} = pp_0$. When a 2-edge pq is encountered then $p\bar{p}$ is the edge pp' that belongs to pq. A symmetric scan is carried out to find the edge pp'' that belongs to pq. The total time, for all vertices p of R, is clearly $O(n^2)$.

Step 3, *returning the solution*. Step 3 is computationally trivial. It takes time $O(n^2)$ since constant time suffices to test whether or not pp', pp'', qq', qq'' satisfy the conditions of step 3. However, it is not trivial to see that the edge pq returned in step 3 is also the shortest expandable 2-edge. The proof of this claim is fairly involved; we refer the reader to [EdTa91] for the full details.

The following theorem summarizes the algorithmic implications of all of this.

MinMax Length Theorem. A minmax length triangulation of a set of n points in $\Re^2$ can be constructed in time $O(n^2)$.

The algorithm that constructs a minmax length triangulation in the claimed amount of time is a combination of the algorithms given in Sections 3, 5.1, 5.3, and 6.

7 Arbitrary Normed Metrics

An open convex region $D \subseteq \Re^2$ that is symmetric with respect to the origin can be used to impose a *norm* on $\Re^2$: for a point $x \in \Re^2$ define $\|x\| = \|x\|_D = \alpha$ if x lies on the boundary of $\alpha D = \{\alpha y \in \Re^2 : y \in D\}$. The norm can then be used to impose a (*normed*) *metric* on $\Re^2$: for two points $x, y \in \Re^2$ define $|xy| = |xy|_D = \|y - x\|_D$. D is the *unit-disk* of the metric and the boundary of D is its *unit-circle*. Notice that the three requirements for a metric are indeed satisfied. First, $|ab| = 0$ iff $a = b$ because $\|x\| = 0$ iff x is the origin. Second, $|ab| = |ba|$ because D is centrally symmetric

and therefore $\|x\| = \|-x\|$. Third, the triangle inequality, $|ac| \leq |ab| + |bc|$, follows from the convexity of D. Examples of normed metrics are the l_p-metrics, for $1 \leq p \leq \infty$, and the so-called A-metric discussed in [WiWW85] for its applications to VLSI.

As mentioned in the introduction, the developments in Sections 2 through 6 are all based on a small number of basic facts, namely the distance relations expressed by the $\square$-Lemma and the $\triangle$-Lemma, the convexity of the lune of an edge, and the straightness of the bisector of two points. Only the last fact is not valid in the case of arbitrary normed metrics; nevertheless, we have shown in the full paper how to resolve this. Thus, we have the following:

General MinMax Length Theorem. Let S be a set of n points in $\Re^2$ equipped with a normed metric with strictly convex unit-disk. Given the relative neighborhood graph, a minmax length triangulation of S can be constructed in time $O(n^2)$.

The result stated in the General MinMax Length Theorem raises the question of how fast $rng(S)$ can be constructed. The trivial algorithm tests all $\binom{n}{2}$ edges, each in time $O(n)$, and therefore takes time $O(n^3)$. Faster algorithms are known for the l_p-metrics where $O(n \log n)$ time suffices (see [JaKY90] and [Lee85]).

8 Discussion

The main contribution of this paper is the first polynomial time algorithm for computing a minmax length triangulation of a set S of n points in $\Re^2$. Given the relative neighborhood graph of S, the algorithm takes time $O(n^2)$. The algorithm works for arbitrary normed metrics. The polynomial time bound follows because the relative neighborhood graph of S can be found in polynomial time. The question remains whether or not a minmax length triangulation can be computed in less than quadratic time.

The results of this paper are an out-growth of our general efforts to understand triangulations that optimize length criteria. There are, however, still many related problems whose complexities remain open. These include the problem of minimizing the entire vector of edge-lengths, the minimum length triangulation problem, and the maxmin length triangulation problem.

References

[ACNS82] M. Ajtai, V. Chvátal, M. M. Newborn and E. Szemerédi. Crossing-free subgraphs. *Ann. Discrete Math.* **12** (1982), 9–12.

[BrZl70] J. Bramble and M. Zlámal. Triangular elments in the finite element method. *Math. Computation* **24** (1970), 809–820.

[Dela34] B. Delaunay. Sur la sphère vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* **7** (1934), 793–800.

[Edel87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, Heidelberg, Germany, 1987.

[EdMü90] H. Edelsbrunner and E. P. Mücke. Simulation of Simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graphics* **9** (1990), 66–104.

[EdTa91] H. Edelsbrunner and T. S. Tan. A quadratic time algorithm for the minmax length triangulation. Techn. Rep. UIUCDCS-R-91-1665, Dept. Comput. Sci., Univ. of Illinois, Urbaba, 1991.

[EdTW90] H. Edelsbrunner, T. S. Tan and R. Waupotitsch. An $O(n^2 \log n)$ time algorithm for the minmax angle triangulation. *In* "Proc. 6th Ann. Sympos. Comput. Geom., 1990", 44–52.

[ElAv81] H. ElGindy and D. Avis. A linear algorithm for computing the visibility polygon from a point. *J. Algorithms* **2** (1981), 186–197.

[GuSt85] L. J. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Trans. Graphics* **4** (1985), 74–123.

[JaKY90] J. W. Jaromczyk, M. Kowaluk and F. F. Yao. An optimal algorithm for constructing β-skeletons in L_p metric. To appear in *SIAM J. Comput.*

[Klin80] G. T. Klincsek. Minimal triangulations of polygonal domains. *Annals Discrete Math.* **9** (1980), 121–123.

[Lank69] P. M. Lankford. Regionalization: theory and alternative algorithms. *Geographical Analysis* **1** (1969), 196–212.

[Laws77] C. L. Lawson. Software for C^1 surface interpolation. In *Math. Software III*, J. R. Rice, ed., Academic Press, 1977, 161–194.

[Lee85] D. T. Lee. Relative neighborhood graphs in the L_1-metric. *Pattern Recognition* **18** (1985), 327–332.

[Ling87] A. Lingas. A new heuristic for minimum weight triangulation. *SIAM J. Algebraic Discrete Methods* **8** (1987), 646–658.

[Lloy77] E. L. Lloyd. On triangulations of a set of points in the plane. *In* "Proc. 18th Ann. IEEE Sympos. Found. Comput. Sci., 1977", 228–240.

[PlHo87] D. A. Plaisted and J. Hong. A heuristic triangulation algorithm. *J. Algorithms* **8** (1987), 405–437.

[PrSh85] F. P. Preparata and M. I. Shamos. *Computational Geometry - an Introduction*. Springer-Verlag, New York, 1985.

[Raja91] V. T. Rajan. Optimality of the Delaunay triangulation in $\Re^d$. *In* "Proc. 7th Ann. Sympos. Comput. Geom., 1991", 357–363.

[Sibs78] R. Sibson. Locally equiangular triangulations. *Comput. J.* **21** (1978), 243–245.

[Supo83] K. J. Supowit. The relative neighborhood graph, with an application to minimum spanning trees. *J. Assoc. Comput. Mach.* **30**, 428–448.

[Tous80] G. T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition* **12** (1980), 261–268.

[WiGS90] F. W. Wilson, R. K. Goodrich and W. Spratte. Lawson's triangulation is nearly optimal for controlling error bounds. *SIAM J. Numer. Anal.* **27** (1990), 190–197.

[Wism80] S. K. Wismath. Triangulations:an algorithmic study. Techn. Rep. 80-106, M. Sc. Thesis, Dept. Comput. Sci., Queen's Univ., Kingston, Ontario, 1980.

[WiWW85] P. Widmayer, Y. F. Wu and C. K. Wong. Distance problems in computational geometry with fixed orientation. *In* "Proc. 1st Ann. Sympos. Comput. Geom., 1985", 186–195.