Refining a Triangulation of a Planar Straight-Line Graph to Eliminate Large Angles

Scott A. Mitchell*

Abstract

We show that any PSLG with v vertices can be triangulated with no angle larger than $7\pi/8$ by adding $O(v^2 \log v)$ Steiner points in $O(v^2 \log^2 v)$ time. We first triangulate the PSLG with an arbitrary constrained triangulation and then refine *that triangulation by adding additional vertices and edges. We follow a lazy strategy of starting from an obtuse angle and exploring the triangulation in search of a sequence of Steiner points that will satisfy a local angle condition. Explorations may either terminate successfully (for example at a triangle vertex), or merge.*

Some PSLGs require $\Omega(v^2)$ Steiner points in any triangulation achieving any largest angle bound less than π. Hence the number of Steiner points added by our algorithm is within a $\log v$ factor of worst case optimal. For most inputs the number of Steiner points and running time would be considerably smaller than in the worst case.

1 Introduction

1.1 Problem statement and motivation

We are concerned with finding a *Steiner triangulation* of an embedded planar straight-line graph (PSLG). That is, we seek an embedded triangular graph, such that vertices of the input appear as vertices of the output, and edges of the input appear as a union of edges of the output. The added vertices of the output are called Steiner points. The triangulation we seek must be *conformal*, that is, two faces of the triangulation must intersect at a face of the triangulation, or not at all. A vertex in the interior of an edge is called *non-conformal*.

**Email:samitch@cs.sandia.gov.* Applied and Numerical Mathematics Department, Mail stop 1422, Sandia National Laboratories, PO Box 5800, Albuquerque, New Mexico 87185-5800. This work performed at Sandia National Laboratories and supported by the U.S. Department of Energy under contract DE-AC04-76DP00789, and by the Applied Mathematical Sciences program, U.S. Department of Energy Research.

Steiner triangulations whose triangles have bounded shape are important for numerical analysis, in particular for a mesh in a finite element method. Babuška and Aziz [1] shows that the convergence of a finite element method depends on the largest angle of the triangulation. Often one wishes to find a triangulation for a PSLG that is not a polygon. For example, a semiconductor may have two differently doped regions. Hence a description of the semiconductor would include an edge with the interior of the region on both sides. Steiner triangulations without large angles are also of use in functional interpolation and computer graphics (see Barnhill [2]).

In addition to the shape of the triangles, another important criterion for a triangulation is the number of triangles. For example, in a finite element method calculation the number of triangles directly affects the running time. For many triangulation algorithms the number of triangles produced depends on the input geometry or embedding, and not just the cardinality of the input. Bern, Dobkin, and Eppstein [3] pose as an open problem the existence of an algorithm to triangulate a PSLG without large angles using only a polynomial number of Steiner points. Here "polynomial" is taken to mean polynomial in the input cardinality, independent of the geometry.

1.2 Previous results

For polygonal input there are many results concerning the construction of triangulations without large angles. Bern and Eppstein [4] shows how to triangulate an arbitrary polygon so that no angle is obtuse by adding $O(v^2)$ Steiner points. Bern, Dobkin, and Eppstein [3] shows how to triangulate various types of input polygons with various angle bounds, using between $O(v \log v)$ and $O(v^{1.85})$ Steiner points. Bern, Eppstein and Gilbert [6] shows how to triangulate a point set with no obtuse angles using only a linear number of Steiner points, which is worst case optimal. Eppstein [9] achieves this and simultaneously approximates the minimum weight Steiner triangulation. Bern, Mitchell and Ruppert [7] has very recently

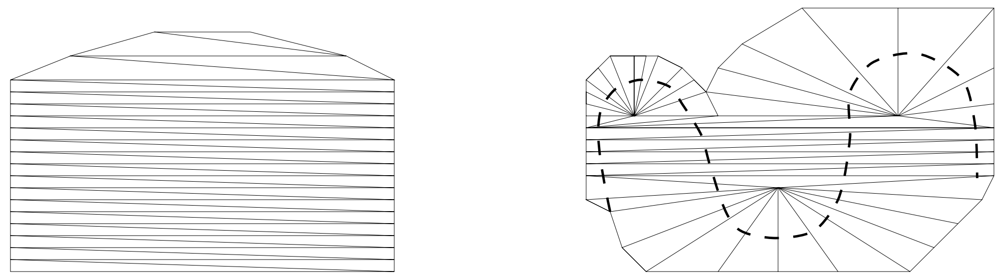

Figure 1: A triangulation refinement may require $\Omega(nm)$ Steiner points to have any constant angle bound (left). In fact, a single Steiner path may require $\Omega(np)$ Steiner points (right).

shown how to triangulate an arbitrary polygon so that no angle is obtuse using only $O(v)$ Steiner points, matching the worst case lower bound.

Ruppert [12] shows how to triangulate a PSLG so that no angle is smaller than $\pi/9$, and hence no angle is larger than $7\pi/9$. However, any triangulation that achieves no small angles is doomed to use a non-polynomial number of Steiner points, dependent on the input geometry. There are several previous algorithms that achieve similar results (by dissimilar techniques) for polygonal input. See Bern and Eppstein [5] for a summary.

Edelsbrunner, Tan, and Waupotitsch [8] shows how to generate a *constrained triangulation* (one where no Steiner points are allowed) of a PSLG such that the maximum angle is minimized. The technique used is edge-insertion, a global strategy that is a generalization of local edge flip. Mitchell [10] shows how to generate a *covering triangulation* (one where no Steiner points are allowed on the input edges) of a PSLG such that the maximum angle is approximately minimum, using a linear number of Steiner points.

Once a triangulation with bounded large (or small) angles has been constructed, one may wish to refine it to reduce the area of triangles in a certain region, while still maintaining an angle bound. Rivara [11] gives an overview of the considerable effort that has been devoted to this problem.

1.3 Overview

We consider the problem of triangulating a PSLG so that no angles are large. We solve this by first triangulating the PSLG with an arbitrary constrained triangulation, and then refining that triangulation. Given any triangulation, we show how to *refine* it by adding additional vertices and edges so that no angle is larger than $7\pi/8$. Our construction adds $O(nm + np \log m)$ vertices and runs in time $O((nm + np \log m) \log(m + p))$. We define p to be one plus the number of holes and interior vertices in the original triangulation. That is, p is the number of one-dimensional connected components of the boundary of the region to be triangulated, plus the number of vertices strictly interior to the region to be triangulated. We define n to be the number of small angles, where small means less than $\pi/8$ (the supplement of our desired angle bound). We define m to be one plus the number of obtuse angles in the original triangulation. By Euler's formula, in any constrained triangulation of a PSLG with v vertices each of p, n and m is $O(v)$. Hence the final PSLG triangulation has $O(v^2 \log v)$ vertices and takes $O(v^2 \log^2 v)$ time. For a triangulation of a simple polygon, we have $p = 1$, so our algorithm adds $O(v^2)$ Steiner points (matching the worst case lower bounds below).

Bern and Eppstein [4] shows how to refine a constrained triangulation of a simple polygon so that no angle is obtuse using $O(v^4)$ Steiner points. They provide a lower bound example, due to Paterson, that illustrates the key concept in our algorithm. The example shows that a triangulation refinement may require $\Omega(v^2)$ (actually $\Omega(nm)$) Steiner points in order to achieve any angle bound less than π. The example consists of a stack of $n = \Omega(v)$ long and skinny triangles capped by $m = \Omega(v)$ triangles with obtuse angles directed into the stack as in Figure 1 left. Each obtuse angle in the cap requires a Steiner point on the opposite triangle edge in order to refine the triangulation without large angles. This induced Steiner point in turn induces a Steiner point on the next lower edge, etc. If the figure is made sufficiently wide and short, the Steiner points induced for different obtuse angles are far apart and can not interact with one another. Hence each of the $\Omega(v)$ obtuse angle induces $\Omega(v)$ Steiner points, for a total of $\Omega(v^2)$ Steiner points.

Steiner path. The key concept in our algorithm is the fact that if the final triangulation is to have no large angles, adding a Steiner point on one edge of a triangle may induce the addition of a Steiner point on another edge of the triangle. We call a sequence of induced Steiner points a *Steiner path.* Besides being fairly intuitive, the fact that Steiner paths are sometimes necessary can be proved as a direct result of a lemma about constrained triangulations in Edelsbrunner, Tan, and Waupotitsch [8] (see Section 2.1).

A variation on Paterson's example provides additional motivation for Steiner paths in Section 2. We can change the direction of propagation of a Steiner path with a sequence of triangles all having a vertex in common: We build the example of Figure 1 right by using $p = \Omega(v)$ such constructions separated by a middle stack of size $n = \Omega(v)$. The Steiner path shown in Figure 1 right is required to intersect each edge of the middle stack $\Omega(p)$ times. Hence a triangulation with a single obtuse angle may require $\Omega(v^2)$ (actually $\Omega(np)$) Steiner points in any refinement that achieves an angle bound less than π.

Algorithm. Our algorithm is as follows. Given a PSLG, we triangulate it with an arbitrary constrained triangulation algorithm, such as the minmax angle triangulation of Edelsbrunner, Tan, and Waupotitsch [8]. Henceforth we consider that triangulation as our input PSLG. For each obtuse angle of the input, we subdivide it into two acute angles by adding the altitude from it to the opposite triangle edge. Hence all triangles are non-obtuse (but also non-conformal), which is important for Section 3. These altitudes introduce a collection of Steiner points in the interior of triangle edges. These non-conformal points induce Steiner paths.

Any fixed strategy of consecutively chosing the Steiner points on a path is doomed to produce a very long path for some input. Instead, we adopt a "lazy" refinement approach: For a given desired angle bound there is some flexibility in picking the next Steiner point on a path (Section 2). We retain this flexibility, and consecutively determine an ever widening region called a *horn*, such that there is some acceptable Steiner path from the initial vertex to every point in the horn. Only later do we chose exactly which path we take from among all those possible inside the horn. This allows us to bound the length of a particular path by $O(np)$.

Eventually each horn will terminate either by intersecting the boundary of the input or a triangle vertex, by intersecting itself in a special way, or by intersecting another horn. In the first case we create a Steiner path to a Steiner vertex on the input boundary, or the triangle vertex. In the second case we create a Steiner path that ends in a loop (see Figure 4). In the third case, we can create a Steiner path to an intersection point of the two horns on a triangle edge (see Figure 6).

Because of the third case our algorithm is iterative: We may have to create a Steiner path for the intersection point, which we do in the next iteration. The number of Steiner paths m_i we need to introduce at iteration i decreases geometrically, so there are $O(\log m)$ iterations. The collection of paths may intersect an input edge $O(m_i + p)$ times, which is surprisingly close to our bound of $O(p)$ for single path. Hence at each iteration we add $O(n(m_i + p))$ Steiner points, for a total of $O(nm + np \log m)$. As a practical consideration, the constants in this bound are relatively small. In particular, we derive an upper bound of $9nm + 24np \log_{3/2} m + 2n/3 + m$ Steiner points, and even this is not tight. Furthermore, assuming most inputs do not have long sequences of adjacent triangles with small angles, for most inputs the refinement algorithm would add considerably fewer points (perhaps only $3m$).

When introducing a path, we just introduce vertices, and not edges between consecutive vertices of the path. We do this because edges for two different paths may cross interior to a triangle. We introduce edges to make the graph conformal only after all paths have been created. We resolve the crossings of Steiner path edges in two ways. If two crossing edges have vertices near a small angle vertex of a triangle, we can swap vertices so that the edges do not cross, and add a diagonal to triangulate the resulting quadrilateral. This strategy does not work near the large angle vertices of a triangle, since paths involving all three of the triangle edges may interact. So instead we introduce one vertex inside the triangle near the edge opposite the small angle, and connect it with an edge to each remaining vertex on the triangle boundary.

The remainder of this paper is organized as follows. Section 2 concerns the development of the Steiner paths, and Section 3 describes how to fix the non-conformal input triangles. In Section 4 we present selected open problems. The omitted proofs may be found in the full paper.

2 Introducing Steiner points

2.1 Steiner path motivation

Consider refining a given triangulation so that no angle is greater than some bound. From Edelsbrunner, Tan, and Waupotitsch [8] we have the following lemma, where $\mu(\mathcal{T})$ denotes the maximum angle of a triangulation $\mathcal{T}$.

Lemma 1 *Given a vertex set A, in any constrained triangulation $\mathcal{T}$ containing edge $\overline{WV}$, we have $\mu(\mathcal{T}) \geq \max_{S \in A} \angle WSV$.*

For a Steiner triangulation, the edge opposite a large angle of a triangle must be subdivided in order to reduce the bound of this lemma (adding Steiner vertices elsewhere merely increases A). So suppose we add a vertex S to subdivide an edge. Unless S is on the boundary of the region to be triangulated, there may be a triangle edge $\overline{VW}$ that subtends a large angle at S. Hence to reduce the bound of the lemma we need to subdivide this edge as well, etc., inducing a Steiner path.

To gain some intuition about long Steiner paths, we have a sufficient condition on a triangle T such that $\angle VSW$ is not large: $\angle VSW$ is at least the supplement of the smallest angle of T. Hence Steiner paths only continue through triangles with a small angle.

If $\angle VSW$ is large, we wish to find an acceptable placement of S_1 on $\overline{WV}$ in terms of the angles $\angle VSS_1$ and $\angle WSS_1$ such that the lower bound from Lemma 1 is reasonably small: If we place S_1 so that both of these angles are less than $\alpha \geq \pi/2$, then the lower bound from Lemma 1 is at most α. Requiring $\angle VSS_1 = \angle WSS_1 = \pi/2$ leads to a single choice for S_1. This is too restrictive in that it could lead to long Steiner paths (e.g. infinitely long in Figure 4). We chose $\alpha = 3\pi/4$, so that there is a range of acceptable placements for S_1.

In fact, regardless of the choice of α, we need a global strategy for placing the S_j that takes into account the entire induced Steiner path: For any local strategy, there is an input that leads to a very long Steiner path.

We do not fix S_1, but instead only consider it to be in the acceptable range, with the freedom to go back and fix its exact location later. Hence we have more freedom in where to place S_2, or we may even discover a position for S_1 where S_2 is unnecessary due to a triangle vertex of T_1. The longer the path is, the more freedom we have in the placement of the last Steiner vertex (see Figure 2). We are able to take advantage of this freedom to prove that paths are only of length $O(np)$. Recall that for Section 3, the first point S is always on the altitude containing the large angle vertex, and we consider that altitude to be as any other triangle edge. We now formalize.

2.2 Horns

Cone. Consider a Steiner point S on $\overline{WU}$ of $\triangle UVW$. The *cone* at S consists of all points P of $\triangle UVW$ such that $\angle PSW$ and $\angle PSU$ are at most $3\pi/4$. Alternatively, the cone is the intersection of $\triangle UVW$ and the infinite sector at S whose bounding rays make angle $\pi/4$ with $\overline{WU}$. The angle between the bounding rays of the cone is $\pi/2$.

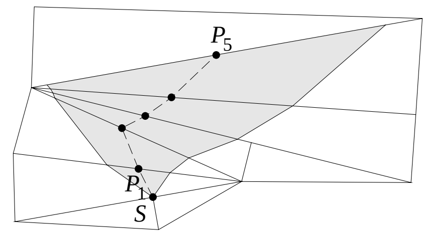

Figure 2: A horn (shaded) and its center path (dashed) terminating on its maw.

Maw. The *maw* of a cone is the portion of the cone on the boundary of $\triangle UVW$, excepting S itself. If triangle vertex V is in the maw, then the lower bound from Lemma 1 is at most $3\pi/4$. Otherwise, the maw is contained in $\overline{WV}$, and corresponds to a range of positions for S_1 ensuring that the lower bound of Lemma 1 is at most $3\pi/4$.

Horn. We iteratively build the *horn* at S as a union of cones. Initially the horn is the cone at S. The horn at stage $j+1$ is the union of cones for the points in the maw at stage j. See Figure 2.

Center and boundary paths. We define the *center path* of a horn to be the sequence of line segments connecting the midpoints of the maws for consecutive stages, starting at S. See Figure 2. We call the two sequences of segments from the starting vertex of a horn making angle $\pi/4$ with each triangle edge *boundary paths.*

Terminating criteria. We continue the construction of the horn in stages until one of the following occurs (the first three are considered case one in the introduction):

1. The maw contains a triangle vertex. As a heuristic, we also terminate if the maw contains a Steiner path vertex of a previous iteration (see Section 2.5).

2. The maw is on a triangle edge where the next triangle has all angles at least $\pi/8$.

3. The maw is on an edge of the boundary of the region to be triangulated.

4. The maw contains a center path point of a previous stage of the horn, and moreover the horn defined from that center path point contains that center path point (see Figure 4).

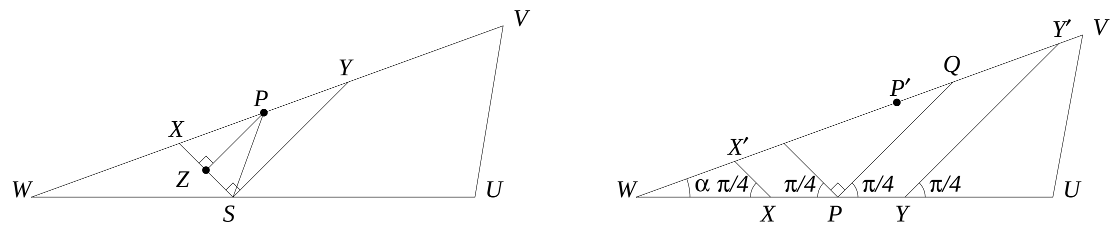

Figure 3: In a cone, the maw is twice the center path (left). In a horn, the maw is more than the center path (right).

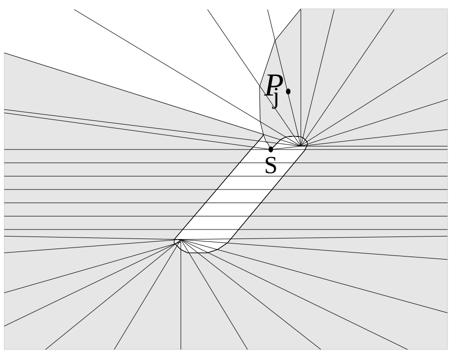

Figure 4: A horn may self intersect (left). The horn shown will self terminate by Item 4 after three more stages, since the horn from center path point P_j will contain P_j.

5. The maw intersects a horn constructed earlier in the current iteration, and either the horns are oriented in the same direction, or in opposite directions and the maw intersects a boundary path or center path of the other horn (see Figure 6).

Acceptable path. When one of these termination criteria occurs, there is an *acceptable* Steiner path inside the horn. By an acceptable path, we mean that the cone from a Steiner path vertex contains the next Steiner path vertex. Except for the last triangle in Item 5, from this it follows that the lower bound from Lemma 1 for all triangles touched by the path is no more than $3\pi/4$ ($3\pi/8$ for the last triangle in Item 2). Given a final Steiner point, it is easy to compute an acceptable path by working from the final Steiner point back to the first point S of the horn. If the horn terminates because of Item 4, then we must form a loop containing the distinguished center path point as well (see Figure 4).

In Item 1, the final Steiner point is the vertex contained in the maw. In Item 2 and Item 3 we may pick any point in the maw. In Item 5, we pick the point that is both in the maw and on the boundary path of the other horn. (We may not actually use the corresponding path, depending on whether any later horns in the current iteration terminate on it; see Section 2.5). In Item 4, we pick the center path point P_j that caused the horn to terminate.

2.3 Bounding a single path

In order to bound the number of times horns may intersect a given edge, we need some lemmas about how quickly the maw of a horn grows in relation to its center path length.

Lemma 2 *For the cone at S, the width of the maw is twice the length of the center path.*

In general, the center point of the next cone is not the center point of the next maw, so that the width of the maw is not always twice the center path length. However, we are able to establish a smaller linear bound.

Lemma 3 *The width of the maw is greater than the length of the center path. Also, the cone for P_j contains P_{j+1}.*

Lemma 4 *If a horn intersects an edge with maw width M and again at a later stage with maw width M', then $M' > 2M$.*

We wish to bound the number of times a center path may cross a given edge. We first show that successive center path points have some ordering along an edge.

Lemma 5 *Suppose a horn intersects an edge E with center path point P_1, and again at a later stage with center path point P_2. If the horn intersect E at a later stage with center point P_3 between P_1 and P_2, then it self terminates as in Item 4.*

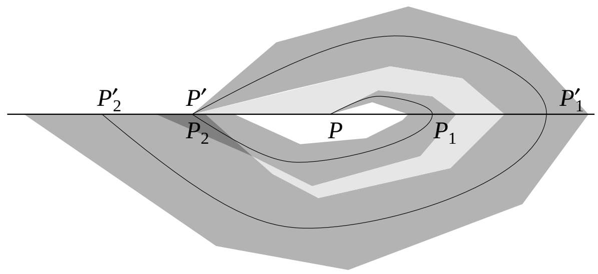

Figure 5: Here we show two consecutive reversals, their horns (darkly shaded), and the inverse horn between them (lightly and very darkly shaded). The inverse horn terminates with negative width, where the two horns overlap (very darkly shaded).

Proof. This proof illustrates a general technique we use often: We show that a center path must be long, so that a maw is wide. Hence either the center path is far away from some feature or the maw contains that feature.

Consider the horn from P_1 (whether or not P_1 really is the first point of a horn). Let d be the distance from P_1 to P_2. Let M_2 be the maw width at P_2, where from Lemma 3 $M_2 > d$. Let M_3 be the maw width at P_3. From Lemma 4 we have $M_3 > 2M_2$. Hence $M_3 > 2d$, and if P_3 has distance to P_1 less than d, then the maw at P_3 contains P_1 and the horn terminates as in Item 4. ∎

We may use this lemma to analyze the number of intersections a center path may have on an edge if all points of intersection lie on one side of the first center path point.

Hop. The center path between two center path points P_j and P_{j+1} on an edge E, together with the portion of E between P_j and P_{j+1}, forms the boundary of a compact set in the plane. We call this set a *hop*.

Theorem 1 *Suppose a horn intersects an edge E with center path point P_1. Consider the line L containing E. Then the horn may intersect E at most p times on one side of P_1 before intersecting L on the other side of P_1.*

Proof. By Lemma 5, $P_2, P_3, \ldots$ are consecutive along E, and hence the corresponding hops have disjoint interiors. Each hop contains at least one vertex of the input in its interior (else the horn crosses an edge twice consecutively, a contradiction) and hence there can be at most p such hops. ∎

We now consider the case that all of the points of a center path on E do not lie on the same side of the first point. For this to happen, there needs to be a *reversal*. Intuitively, a reversal is two consecutive hops that travel in opposite directions.

Reversal. A reversal is the center path of a horn from a point P on edge E to a point P_1 on L (or E) to a point P_2 on E, where P_1 and P_2 are on opposite sides of P, and L is the line through E. Also, the center path must not cross E at any point other than P_1 between P and P_2 (otherwise we may find a shorter reversal instead). See Figure 5. Just as for a hop, we say that a reversal contains the input vertices in the region bounded by the center path from P to P_2 and E.

Lemma 6 *The vertices contained in two reversals of a center path are not identical, unless the horn terminates as in Item 4.*

Proof. The proof lies in the observation that if hops contain the same vertices and are oriented in the same direction, then they grow closer together as their stages increase. This holds true for two hops for the same Steiner path but different starting stages, and also for two hops for different Steiner paths (used in the next subsection).

We define an *inverse horn* as the region between the hops of two such horns, and its maw width is the distance between the two horns. See Figure 5. The rate of decrease of the inverse horn width can be bounded below in the same way that the rate of increase of the width of a maw can be bounded below. Hence the proof reduces to showing that the center path of the inverse horn is long, so that the outer reversal must contain the starting center path vertex of the inner reversal. ∎

Theorem 2 *A horn may intersect a given edge at most $O(p)$ times.*

Proof. Because of reversals, hops are not necessarily disjoint. However, hops are partially ordered by containment. Hence there are at most $2p$ unique input vertex sets contained in hops. We enumerate the hops by charging the vertex sets for hops. Using a careful charging scheme, Lemma 6, and Theorem 1, each vertex set gets charged at most four times. ∎

2.4 Bounding the collection of paths

We now consider all of the horns in a given iteration, and how they interact. We seek a bound on the number of times these horns may collectively cross a given edge. We consider two horns with hops oriented

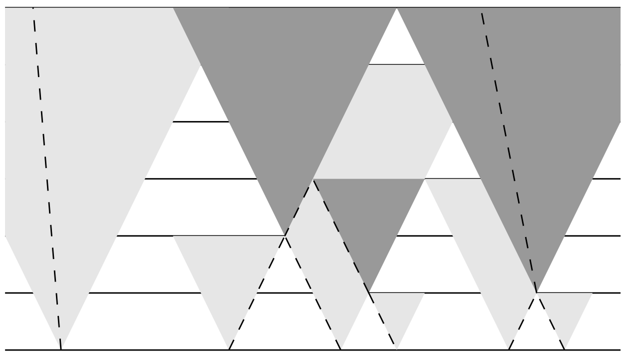

Figure 6: A collection of horns may merge. The horns for the next iteration are darkly shaded, and Steiner paths are dashed.

in the same direction with respect to E. In the following two lemmas we show that two horns cannot each have two consecutive hops containing the same triangle vertices.

Lemma 7 *Suppose a horn H has two consecutive hops M and M_1 (not a reversal), and another horn H' has two consecutive hops M' and M'_1, such that M and M' contain the same input vertex set, and similarly M_1 and M'_1 contain the same input vertex set. If neither H nor H' self terminates on M, M', M_1 or M'_1 as in Item 4, then M_1 and M'_1 intersect as in Item 5.*

Lemma 8 *Suppose a horn H has a reversal R, and another horn H' has a reversal R' with hops that contain the same vertex sets as those of R. If neither horn terminates on R or R' as in Item 4, then the two reversals intersect as in Item 5.*

Theorem 3 *The collection of horns may intersect a given edge $O(m_i + p)$ times.*

Proof. The proof is very similar to the proof of Theorem 2, and relies on on Theorem 2, Lemma 7 and Lemma 8. ∎

2.5 Introducing Steiner paths for the collection

The algorithm for constructing the Steiner paths is iterative. If no horn terminated by Item 5 then we could construct the Steiner paths and no more iterations would be required. However, if a horn terminates because of Item 5 then its last Steiner point may induce a path in the next iteration. We have shown above that each iteration will produce only $O(n(m_i + p))$ Steiner path points, where m_i is the number of horns in iteration i. We show below that at each iteration we reduce the number of horns by at least a factor of 2/3, so the sum of the m_i is $3m$. Furthermore, since the m_i are bounded above by a geometric series, there is only a logarithmic number of iterations in terms of m.

Theorem 4 *The number of horns in the next iteration is at most 2/3 times the number of horns in the current iteration. That is, $m_{i+1} \leq 2m_i/3$.*

Proof. The proof of this theorem is a matter of specifying exactly which paths we create when horns intersect. We may consider a horn H as a root of a subtree, whose children are the like-oriented horns that terminate on H as in Item 5. The children are divided into left and right sides, depending on where they intersect H. (There are also the oppositely oriented horns that terminate on the center or boundary path of H, but such horns will not contribute to the next iteration.)

In any tree, there is a node N whose children are all leaves. We may recursively remove the subtree at N by adding a tree-like Steiner path for each non-empty side of N. The "root" vertex of such a Steiner path lies on the boundary path of the corresponding horn, and gives rise to a horn in the next iteration. See Figure 6. As a heuristic, it is sometimes worthwhile to continue a Steiner path along the boundary path to the parent of N, and sometimes worthwhile to introduce no path on a side with one child. If there is one child on each side, then we remove a subtree of size three and add two root vertices. In all other cases, the ratio of the size of the removed subtree to the number of added root vertices is at least 2. Hence $m_{i+1} \leq 2m_i/3$.

Note that the root of the entire tree terminates by a different criteria, and admits a Steiner path that does not give rise to a horn in the next iteration. Thus subtrees of size one need not be accounted for. ∎

Theorem 5 *At most $O(nm + np \log m)$ Steiner points are added.*

3 Triangulating the non-conformal triangles

We now have a non-conformal triangulation in which some triangles have Steiner points on their edges. We now show how to triangulate these triangles, taking advantage of the special geometry of the Steiner points in order to ensure that all angles are at most $7\pi/8$.

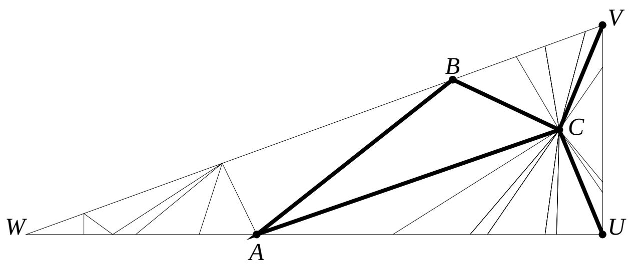

Figure 7: How to triangulate a triangle with Steiner points on its boundary.

3.1 Fixing small angle triangles

We introduce the Steiner path edge $\overline{AB}$ when the angle between the triangle edges containing A and B is less than $\pi/8$. We say such edges are *drawn*. The other Steiner path edges are forever ignored, and also some drawn edges may be erased later. We have bounds on the angle that Steiner path edges make with triangle edges.

Lemma 9 *The angles at the intersection of a drawn Steiner path edge and a triangle edge are between $\pi/8$ and $7\pi/8$.*

We can swap vertices of drawn Steiner path edges so that they do not cross, and still maintain the above angle bounds.

Lemma 10 *Steiner path edges may be redrawn so that no two cross, while maintaining the angle bounds of Lemma 9.*

We retain only the drawn edges that are far from $\overline{UV}$: We erase all Steiner path edges $\overline{AB}$ where both $\angle UVA \leq \pi/4$ and $\angle VUB \leq \pi/4$. Since the edges do not cross, there will be an edge $\overline{AB}$ "closest" to $\overline{UV}$ that is not erased: Any edge with a vertex on $\overline{AU}$ or $\overline{BV}$ will be erased, and any edge with a vertex on $\overline{AW}$ or $\overline{BW}$ will be drawn.

What remains is a the non-conformal trapezoid $ABVU$, and conformal triangles and trapezoids bounded by drawn edges, $\overline{UW}$, and $\overline{VW}$. According to Lemma 9 and Lemma 10, the triangles and trapezoids have largest angle no more than $7\pi/8$. Hence the conformal trapezoids may be triangulated with an arbitrary diagonal and achieve largest angle no more than $7\pi/8$.

We now triangulated region $ABVU$. We introduce a vertex C in the interior of the triangle. We place C so that $\angle UVC = \angle VUC = \pi/8$. We triangulate by introducing an edge from C to each of the Steiner and triangle vertices in region $ABVU$. See Figure 7.

It may be that $B = V$ or $A = U$. Also there may not be a drawn edge $\overline{AB}$, so that the region degenerates to $\triangle WVU$. The construction needs no modifications for these cases.

Lemma 11 *The triangulation of region $ABVU$ has no angle larger than $7\pi/8$.*

3.2 Fixing all-large angle triangles

It remains to consider triangles with every angle larger than $\pi/8$. Using a construction almost identical to that of Section 3.1, we may triangulate with no angle larger than $7\pi/8$ (see Figure 7). It is possible to use the fact that the angle at W is large to directly bound the largest angle in the region WAB. For region $ABVU$, the proof of Lemma 11 holds with slightly different angle bounds in various places, but with the same overall bound of $7\pi/8$.

4 Conclusions

There is a tradeoff between the cone angle of the horns and the number of times that a triangle edge is crossed. We state our cone angle to be $\pi/2$. If the cone angle is ϕ, we conjecture that the techniques of Section 3 can be used to obtain triangles with largest angle at most $3\pi/4 + \phi/4$. On a more general note, we have the following problems.

What is the relationship between the largest angle permitted in a triangulation and the number of Steiner points necessary to achieve that bound? How does this depend on the type of input (e.g. convex polygon, simple polygon, polygon with holes, PSLG)?

The cardinality of our PSLG triangulations is within a $\log v$ factor of worst case optimal. Is there an algorithm that is within a constant factor of worst case optimal? A more interesting open problem is the existence of an algorithm that generates triangulations of PSLGs or polygons with cardinality within a factor of optimal for the given input.

References

[1] I. Babuška and A. K. Aziz, On the angle condition in the finite element method, *SIAM J. Num. Anal.* 13:214–226, 1976.

[2] R. Barnhill, Computer aided surface representation and design, *Surfaces in CAGD,* R. Barnhill and W. Boehm, eds., North-Holland, 1–24, 1983.

[3] M. Bern, D. Dobkin, and D. Eppstein, Triangulating polygons without large angles, submitted to *International Journal of Computational Geometry & Applications,* 1991.

[4] M. Bern and D. Eppstein, Polynomial-Size Nonobtuse Triangulations of Polygons, *7th ACM Symposium on Computational Geometry,* pp. 342–350, 1991.

[5] M. Bern and D. Eppstein, Mesh generation and optimal triangulation, *Computing in Euclidean Geometry,* D. Du, and F. Hwang, eds., World Scientific, 1992.

[6] M. Bern, D. Eppstein, J. Gilbert, Provably Good Mesh Generation, *Proc. 31st IEEE Symposium on Foundations of Computer Science,* 231–241, 1990.

[7] M. Bern, S. A. Mitchell and J. Ruppert, Linear-Size Non-Obtuse Triangulation of Polygons, manuscript, 1993.

[8] H. Edelsbrunner, T.S. Tan, and R. Waupotitsch, An $O(n^2 \log n)$ Time Algorithm for the MinMax Angle Triangulation, *Proc. 6th ACM Symposium on Computational Geometry,* 44–52, 1990.

[9] D. Eppstein, Approximating the Minimum Weight Triangulation, *3rd ACM-SIAM Symp. on Discrete Algorithms,* 48–57, 1992.

[10] S. A. Mitchell, Finding a Covering Triangulation Whose Maximum Angle is Provably Small, manuscript, Sandia National Laboratories, 1993.

[11] M.-C. Rivara, A Discussion on the Triangulation Refinement Problem, *Proc. Fifth Canadian Conference on Computational Geometry.* 42–47, 1993.

[12] J. Ruppert, A New and Simple Algorithm for Quality 2-Dimensional Mesh Generation, *Proc. 4th ACM-SIAM Symp. on Discrete Algorithms,* 83–92, 1993.