International Journal of Computational Geometry & Applications

TRIANGULATING POLYGONS WITHOUT LARGE ANGLES

MARSHALL BERN* DAVID DOBKIN† DAVID EPPSTEIN‡

*Xerox Palo Alto Research Center, 3333 Coyote Hill Road
Palo Alto, California 94304, U.S.A.

†Computer Science Department, Princeton University
Princeton, New Jersey 08544, U.S.A.

‡Department of Information and Computer Science, University of California
Irvine, California 92717, U.S.A.

Received (received date)
Revised (revised date)
Communicated by Editor's name

ABSTRACT

We show how to triangulate polygonal regions—adding extra vertices as necessary—with triangles of guaranteed quality. Using only $O(n)$ triangles, we can guarantee that the smallest height (shortest dimension) of a triangle in a triangulation of an n-vertex polygon (with holes) is a constant fraction of the largest possible. For simple polygons, using $O(n \log n)$ triangles, we can guarantee that the largest angle is no greater than 150°. This bound increases to $O(n^{3/2})$ triangles for the case of polygons with holes. We can add the guarantee on smallest height to these no-large-angle results, without increasing the asymptotic complexity of the triangulation. Finally we give a nonobtuse triangulation algorithm for convex polygons that uses $O(n^{1.85})$ triangles.

Keywords: Computational geometry, mesh generation, triangulation, angle condition.

1. Introduction

There have been a number of recent papers on the general problem of triangulating a planar point set or polygonal region with triangles of "guaranteed quality", a problem primarily motivated by mesh generation for finite element methods.[1] The literature on guaranteed-quality triangulation can be initially partitioned according to whether or not extra vertices, or "Steiner points", are allowed.

If Steiner points are not allowed, then the vertices of the triangulation must be exactly the vertices of the input. In this case, computational geometers typically seek a triangulation that exactly optimizes some quality measure. Polynomial-time

†Work partially supported by NSF Grant CCR90-02352.

algorithms have been found for maxmin angle, minmax angle, minmax enclosing circle, maxmin height, and minimizing the longest edge. For point set input that includes elevations for each point, there are algorithms for least-energy and least-slope interpolating terrains. Maxmin angle, minmax enclosing circle, and least-energy terrain are all solved by the well-known Delaunay triangulation.[1]

Allowing Steiner points improves the quality of the optimal triangulation, but forces us to seek only an approximate solution. Steiner points also add another parameter to the quality of the output: the number of triangles in the triangulation, which we call the *size* of the triangulation. Size directly affects computation times in finite element methods, or expense in measuring a continuous function (say concentration of a chemical in groundwater) at sample points in a domain.

Previous papers have shown how to compute a Steiner triangulation with no small angles,[2] no obtuse angles,[3] no small and no obtuse angles,[4,5] and, for point set input, approximately minimum total length.[6] The algorithms for no-small-angle triangulation have size $O(n \log A)$ for point sets and $O(nA)$ for polygons,[2] where A is the maximum aspect ratio (the ratio of the longest side to the height) of a triangle in a (constrained) Delaunay triangulation of the input. These bounds are tight, since there exist inputs that require this many triangles. The algorithms for nonobtuse triangulation use $O(n)$ triangles for point sets[2] and $O(n^2)$ for polygons,[3] but there are no nontrivial lower bounds.

In this paper, we consider some more Steiner triangulation problems. The quality measures considered here are maximizing the minimum height of a triangle and minimizing the maximum angle of a triangle. In the first case, our goal is a triangulation with minimum height at least a constant fraction of the maximum achievable. In the second case, there are two reasonable goals: bounding all angles below 180° by some fixed amount, or requiring all angles to be at most 90°.

What is the motivation? The first problem arises in bounded-aspect-ratio tetrahedralization of polyhedra.[7] Imagine a tetrahedralized polyhedron with a sharp point at vertex v. Center a sphere on v, with radius less than the lengths of the edges incident to v, so that the sphere intersects the polyhedron in a nearly planar, simple polygon P and the interior of the sphere contains no vertices other than v. The tetrahedralization induces a Steiner triangulation of P on the small sphere, and since the interior solid angle at v is small, the aspect ratios of the tetrahedra intersecting the sphere are roughly proportional to the heights of triangles on P. Thus, a tetrahedralization containing only bounded-aspect-ratio tetrahedra, such as that given by Mitchell and Vavasis,[7] necessarily induces an approximately-optimal maxmin-height triangulation of P. Mitchell and Vavasis solve the maxmin-height problem implicitly using octrees, obtaining size something like $O(nA)$; we give an explicit linear-size solution.

Bounding the largest angle below 180° arises in both interpolation[8,9,10] and finite element methods.[11] In particular, Babuška and Aziz[11] proved that finite element error (for a typical differential equation) converges to zero as triangle sizes shrink, so long as the maximum angle in the mesh does not approach 180°. The suprising aspect of this result is that it is not necessary to bound angles away from 0°.

Bounding the largest angle at 90° has special importance. Any smaller fixed bound on the maximum angle necessitates $\Omega(nA)$ triangles for some inputs. A bound of 90° guarantees some desirable numerical properties for finite element methods.[4] Maximum angle 90° also implies the following nice geometric property: all (closed) triangles in a nonobtuse mesh contain their circumcenters; linking circumcenters gives a planar embedding of the planar dual such that dual edges cross at right angles. Finally, nonobtuse triangulation has arisen in computational learning theory.[12]

All of the algorithms given in this paper initially form a grid of horizontal and vertical lines, and then slim down the grid through various techniques. These techniques may be of independent interest.

The following table summarizes the state of our knowledge for four criteria and four increasingly general types of input. Entries indicate the number of triangles needed to achieve the stated criterion; all bounds are given in this paper except the $O(n^2)$ bounds on nonobtuse triangulation[3] and the $O(n^2 \log n)$ bound on no-large-angle triangulation of a planar straight-line graph (PSLG).[13] (The earlier, conference version of this paper[14] mistakenly claimed that our algorithms extended to PSLGs. Actually, a subsequent algorithm due to Mitchell[13] is the first to handle the case of a PSLG with a polynomial number of triangles; his algorithm achieves maximum angle 157.5°.) For the special case of a triangulated simple polygon, Bern and Eppstein[3] have shown an $O(n^4)$ upper bound on nonobtuse triangulation.

	Convex	Simple poly.	Poly. w. holes	PSLG
Maxmin height	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	?
Angles $\leq 150°$	$O(n \log n)$	$O(n \log n)$	$O(n^{3/2})$	$\Omega(n^2)$, $O(n^2 \log n)$
Both of above	$O(n \log n)$	$O(n \log n)$	$O(n^{3/2})$	Impossible
Angles $\leq 90°$	$O(n^{1.85})$	$O(n^2)$	$O(n^2)$	?

The remainder of this paper is organized as follows. Section 2 gives a linear-size algorithm for maxmin height. Section 3 modifies this algorithm to solve the no-large-angle problem while maintaining maxmin height. Section 4 establishes lower bounds for the case of planar straight-line graphs. Section 5 gives the $O(n^{1.85})$ algorithm for nonobtuse triangulation of convex polygons. Finally, Section 6 lists some open problems.

2. Maximizing Minimum Height

We start with the problem of maximizing the minimum height of a triangle. For point set input, the nonobtuse triangulation algorithm of Bern, Eppstein, and Gilbert[2] gives a constant-factor approximation algorithm with $O(n)$ triangles, although we did not point this out in the original paper. For polygon input, however, there are no previous algorithms for this problem. We give a construction for the case of a polygon with holes.

A *polygon with holes* is a region P in the plane whose boundary ∂P consists of a number of disjoint simple polygons. We denote by n the total number of vertices on ∂P. To avoid some degenerate cases, we assume that no edges of ∂P are exactly horizontal or vertical. A triangulation is a partition of P into triangles that intersect only at shared edges and vertices. The *height* of a triangle is the Euclidean distance from its longest edge to the opposite vertex. The height of triangulation $\mathcal{T}$ is the minimum height of a triangle in $\mathcal{T}$.

Definition 1. *The* **feature size** *s of P is the minimum distance, within P, from an edge of ∂P to a vertex of ∂P not on that edge.*

In other words, the feature size is the length of a shortest line segment e joining a vertex of ∂P to an non-incident edge such that $e \cap \partial P$ is either e or the endpoints of e. In any triangulation of P, segment e must cross a triangle with height at most s, so s is an upper bound on the minimum height.

In this section, we show that it is possible to find a Steiner triangulation with height a constant fraction of s, and hence only a constant factor smaller than optimal. Disallowing Steiner points, the optimal height may be much worse; we give two bad examples due to Mitchell.[15] First consider the case that ∂P is simply a regular n-gon with side length s. Any triangulation without Steiner points must contain a triangle with two ajacent sides of ∂P, and hence height $s \cdot \sin(\pi/n)$. A single Steiner point at the center of the n-gon, however, gives height about s. A different sort of bad example is given by a thin rectangle, with two subdivision points near the middle of one long side, say with vertices at $(0,0)$, $(0,1)$, $(2k,1)$, $(2k,0)$, $(k+1,0)$, and $(k,0)$. (A *subdivision point* is a vertex of a polygon at which the angle measures $180°$.) In this example, the difference between Steiner and non-Steiner triangulations depends on k, rather than n. Though all edges necessarily measure at least s in a non-Steiner triangulation, arbitrarily large angles allow much smaller height. Notice that if the sides of a triangle measure $\Omega(s)$ and its angles are bounded away from $180°$, then its height is indeed $\Omega(s)$.

The first step of our maxmin-height algorithm is to "lop off" acute vertices of P. In parallel, we cut off an isosceles triangle at each vertex v where the interior angle measures less than $90°$. The cutting line ℓ_v for acute vertex v crosses the segment joining v to the closest distinct vertex w (by geodesic distance within P) one-third of the way from v to w. We denote by P' the polygonal region remaining after all such cuts. Again assume that P' contains no horizontal or vertical edges.

Lemma 1. *The feature size of P' is at least $s/3$, and each isosceles triangle has feature size at least $s\sqrt{2}/6$.*

Proof. Consider a single vertex v. Cutting line ℓ_v must have length at least $s/3$ because the line segment through w, parallel to ℓ_v and subtended by the angle at v, must have length at least s. The new edge ℓ_v lies at least $s/3$ from non-incident edges in P', because a line segment connecting v to a non-incident edge in P loses at most one-third of its length from each end. Thus, P' has feature size at least $s/3$. The isosceles triangle cut off at v has all acute angles and shortest edge length at least $s/3$. This implies that its height is at least $s\sqrt{2}/6$. $\square$

When we triangulate the remaining polygonal region P', we may introduce Steiner points $\Omega(s)$ apart along the cut lines, which are the bases of the isosceles triangles. As a final step in the algorithm, each isosceles triangle with Steiner

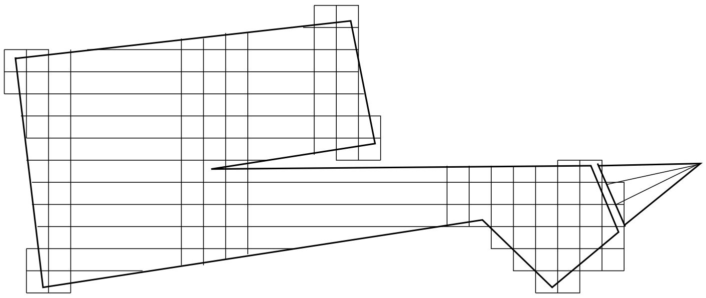

Figure 1. Cutting off acute vertices and dicing.

points on its base is cut into a fan of triangles with angles smaller than $135°$ and height $\Omega(s)$, as shown in Figure 1.

The second step is the *dicing* step, which cuts P' with vertical and horizontal lines. Let $b = s/8$. Imagine the plane divided into an infinite square grid with spacing b. Each vertex of P' falls into a different square in this grid; call these squares *occupied*. Call a square orthogonally or diagonally adjacent to an occupied square a *neighbor* square. Erase all lines of the grid, except ones that support occupied and neighbor squares. Now there is a grid of rectangles. A *segment* is a piece of a grid line inside P' between successive crossings of $\partial P'$. Now erase all parts of remaining grid lines, except the boundaries of occupied and neighbor squares that intersect P' and the segments that support occupied and neighbor squares. There are $O(n)$ faces around $\partial P'$ and $O(n^2)$ interior rectangles. See Figure 1 again.

If a vertex of P' is close in the plane—though geodesically distant—to a non-incident edge of $\partial P'$, then an occupied or neighbor square may intersect P' in two connected components. We treat these anomalies with *Riemann sheets*, that is, we make two copies of such a square and lift one out of the plane in order to embed P'. A Riemann sheet is glued to the plane along the appropriate sides of the square.

The third step is to *warp* some segments of the horizontal and vertical cutting lines to conform to $\partial P'$. Call a point at which $\partial P'$ intersects a vertical or horizontal cutting line a *q-vertex*. In the following two substeps, we do not consider q-vertices to be endpoints of grid edges, so edges are not free to bend at q-vertices. (We can think of a q-vertex on a warping edge as sliding along the edge.)

- Each vertex v of $\partial P'$ chooses the closest vertex c_v (interior or exterior to P') of the square it occupies. The four grid edges meeting at c_v are then replaced by four edges meeting at v, with the other endpoints the same as before. (Our choice of $b = s/8$ guarantees that at most one vertex of each face warps in this step.)

- Each q-vertex that is closer than $b/8$ to a grid vertex chooses its closest grid vertex. The four grid edges meeting at a chosen grid vertex are then replaced by edges meeting at its closest q-vertex. If two q-vertices choose the same grid vertex, then the grid vertex moves to the closer of the two. (Using Riemann

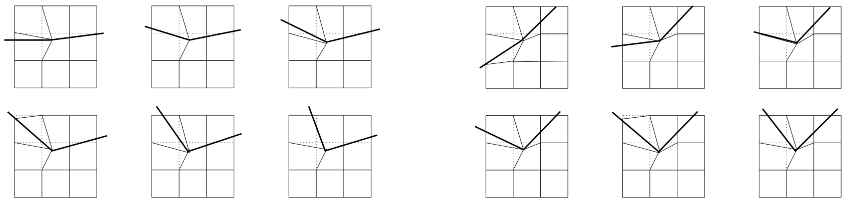

Figure 2. Part of the warping case analysis.

sheets, it is not possible for more than two q-vertices to choose the same grid vertex.)

Now let *edge* mean a maximal line segment in our construction without an embedded q-vertex or grid vertex. All edges have length at least $b/8$, because a smaller edge would be contracted by the second warping substep. Except in neighbor and occupied squares, edges not lying along ∂P remain nearly horizontal and nearly vertical. More precisely, each grid vertex, not part of a neighbor or occupied square, is displaced at most $b/8$ from its original position. If one endpoint of an originally horizontal edge e is displaced up by $b/8$ and the other endpoint displaced down $b/8$, then because e's original length was at least b, e now forms an angle with the horizontal of at most $\arctan((b/4)/b) < 14.04°$. A similar argument shows that originally vertical grid edges deviate from the vertical by at most $14.04°$, except in neighbor and occupied squares.

Lemma 2. *After warping, each face can be triangulated with triangles of height $\Omega(s)$. Neighbor and occupied squares can be triangulated with maximum angle at most $135°$ as well.*

Proof. Assume we have not yet warped, and look at the nine squares around a vertex v of P'. Two edges of $\partial P'$ meet at v. Rotating and reflecting P', we may assume that v lies in the upper left quadrant of the occupied square, that one of the edges crosses the right side or top side, and the angle counterclockwise between this edge and the other measures between $90°$ and $180°$. We conceptually subdivide the sides of the nine squares around v into eight sections of length $b/8$. The warping algorithm's decisions depend only on which sections are crossed, not on where within the section the crossing occurs. Thus we can analyze the warping behavior by considering all possible cases. Figure 2 illustrates part of the tedious, but straightforward, case analysis. The six pictures on the left show all the cases in which one edge of $\partial P'$ crosses the top section of the right side of the occupied square and within $b/8$ of the lower right corner of the top right neighbor square. The six on the right show all the cases in which one edge of $\partial P'$ crosses the top side of the occupied square and the top side of the top right neighbor square, more than $b/8$ from all endpoints. In all cases, all angles are bounded by $135°$.

Now consider a rectangle that is not an occupied or neighbor square. Such a rectangle is crossed by at most two edges of $\partial P'$. Hence a face resulting from such a rectangle has at most six sides. When the face warps, at most two of its vertices move, moving at most $b/8$. Warping leaves the face convex since we do not bend at

q-vertices. At most $m-4$ angles in an m-sided face, $4 \leq m \leq 6$, are larger than $150°$, so we can triangulate by successively cutting off "ears" (triangles with two adjacent sides along the boundary of the face) at which the corner angle measures less than $150°$. Only the last $m-4$ triangles in such an ear triangulation have angles greater than $150°$, and all triangles have height $\Omega(b)$. (A detailed case analysis shows that all faces can actually be triangulated with height almost $b/8$.)

Notice that arbitrarily large angles (such as those in the last $m-4$ triangles) cannot be avoided in some boundary faces, for example, where a nearly horizontal edge of $\partial P'$ crosses a long, skinny rectangle, as along the long bottom edge in Figure 1. □

At this point, we have a triangulation with good height, with $O(n)$ triangles intersecting $\partial P'$ and $O(n^2)$ rectangles (split by diagonals) interior to P'. We now merge rectangles to reduce the size of the triangulation. Erasing each rectangle diagonal and each edge bordering two interior rectangles leaves a set of rectilinear polygons, possibly with holes, nonintersecting except at shared vertices. (A *rectilinear* polygon has boundary that consists of horizontal and vertical segments.) Each such polygon has feature size at least b and satisfies the following condition.

Definition 2. *A rectilinear polygon Q satisfies the* **matching condition** *if every horizontal or vertical line extended from a vertex of Q across the interior first intersects Q at another vertex.*

Lemma 3. *Assume Q is an n-vertex rectilinear polygon (with holes), satisfying the matching condition, and with feature size at least b. Then Q can be triangulated with $O(n)$ triangles of height at least b, adding Steiner points only to the interior of Q.*

Proof. First draw in all interior vertical line segments between vertices of Q, as shown in Figure 3. We treat these verticals in order from left to right. We horizontally project every other vertex from the leftmost vertical line onto the next vertical line. Hence the vertical line numbered 2 in Figure 3 inherits the topmost vertex from line 1, the third from topmost, and so forth. Vertical line 3 inherits every other vertex from line 2. We treat subsequent vertical lines (new Steiner points not pictured) in the same way. Of course, each vertical line also has vertices where it crosses ∂Q, whether or not these vertices are projected onto it from the left, for example, the second vertex from the top on line 2. We add horizontal edges between corresponding vertices (same y-coordinate) on adjacent verticals.

After this left-to-right sweep, we perform a similar sweep starting from the rightmost vertical line. In this sweep, every other vertex on line 10 would be projected onto line 9, and so forth. After the right-to-left sweep each face is a rectangle with minimum feature size at least b and at most one subdivision point on each of its vertical sides. Such a face can be triangulated with triangles of height at least b by adding diagonals in a zig-zag fashion from top to bottom.

The total number of vertices added during the left-to-right sweep is at most $2n$, because the number of projected vertices on vertical line i is at most one more than half the total number of vertices on vertical line $i-1$. A similar observation applies to the right-to-left sweep, giving an overall linear bound on the triangulation. □

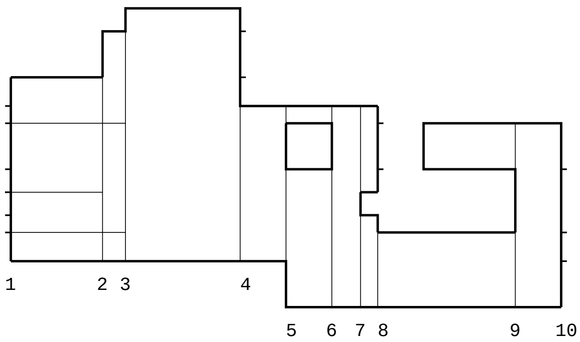

Figure 3. A step in the sweepline algorithm for triangulating rectilinear polygon Q.

Theorem 1. *In time $O(n \log n)$, an n-vertex polygon with holes can be triangulated into $O(n)$ triangles of height $\Omega(s)$, where s is an upper bound on the largest possible minimum height.*

Proof. The algorithm and the bounds on height and number of triangles have been given above; here we prove $O(n \log n)$ running time. We can initially compute feature size s in time $O(n \log n)$ using the Voronoi diagram of the ∂P edges.[16] Cutting off acute vertices involves finding the closest visible neighbor of vertices of ∂P; this computation also takes time $O(n \log n)$ using the bounded Voronoi diagram.[17] Dicing can be accomplished implicitly, that is, we compute the horizontal and vertical lines supporting occupied and neighbor squares, but we do not compute all intersections of these lines. Once boundary faces are sorted around each connected component of $\partial P'$, warping takes only linear time, as there are only $O(1)$ vertices to consider in each face that warps. Removing boundary faces to give rectilinear polygon Q also takes only linear time. Finally we can perform the sweep algorithm in time $O(n \log n)$. $\square$

A practical implementation of the algorithm of Theorem 1 would dispense with Voronoi diagrams. Such an implementation would cut off acute vertices only approximately, not exactly, one-third of the way to the nearest vertex.

3. No Large Angle

In this section, we consider the problem of triangulating the input domain so that all angles are bounded below $180°$. Our algorithms achieve maximum angle $150°$ while (somewhat fortuitously) maintaining minimum height $\Omega(s)$. We require two major modifications of the maxmin-height algorithm given above. The first modification triangulates faces along the boundary $\partial P'$ without large angles. The second modification triangulates the interior rectilinear polygon Q without large angles. This second modification differs for simple polygons and polygons with holes, and hence we obtain different results for triangulation size: $O(n \log n)$ for simple polygons and $O(n^{3/2})$ for polygons with holes.

Let P be a region of the plane whose boundary ∂P is a set of disjoint simple polygons and let s be the feature size of P as before. We lop off all acute corners

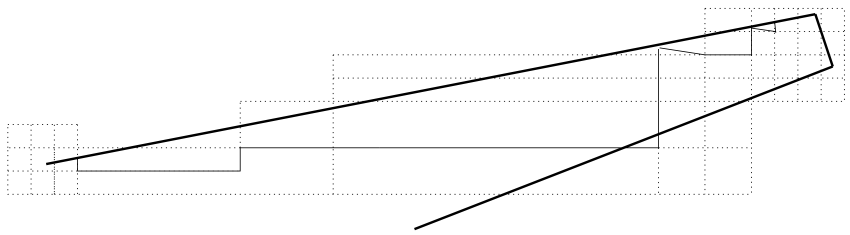

Figure 4. Merged boundary faces along edge e.

(obtaining P'), dice, and warp exactly as in Section 2. As shown in Lemma 2, the isosceles triangles and the faces resulting from occupied and neighbor squares can be triangulated with maximum angle 135°.

We now show how to handle the other faces along $\partial P'$. Our previous construction formed some arbitrarily large angles in these faces, such as along the long bottom edge in Figure 1. Recall that all edges, except those in neighbor and occupied squares and those lying along input edges, deviate from horizontal or vertical by at most 14.04°. Call an edge that forms an angle of at most 14.04° with horizontal (vertical) an *almost-horizontal* (an *almost-vertical*).

Consider an edge e of $\partial P'$ with positive slope. At this point, if there are any angles larger than 150° along e, they occur where a grid edge hits e. Merge all faces supported by edge e, except the faces resulting from occupied and neighbor squares at the endpoints of e. In Figure 4 the faces along edge e (the long top edge) give three merged faces, that meet at points where a grid vertex warped to e. Dotted lines in these faces indicate removed edges. We denote by M a merged face resulting from the steps just described.

Lemma 4. *Polygon M is monotone with respect to a line of slope 1 (that is, any line of slope -1 intersects M in at most one interval). The boundary of M consists of e and a "staircase" of edges that are either almost-horizontal, almost-vertical, or lie along edges of $\partial P'$ with positive slope.*

Proof. First consider the case that all faces supported by e are not supported by other edges of $\partial P'$. Then the boundary of M consists of e, a chain of almost-horizontal and almost-vertical lines that goes upward as it goes to the right, and at most one edge at each end joining e to the chain. It is clear that M is monotone with respect to a line of slope 1.

Now imagine adding other edges of $\partial P'$. These edges can only "clip" portions of the chain from M as shown in Figure 4. No face (except for occupied and neighbor faces) can be supported by two $\partial P'$ edges, one of positive and one of negative slope, since then there would be a vertex inducing a vertical between the two edges. □

Lemma 5. *Let M be an m-vertex monotone polygon as above. M can be triangulated into $O(m)$ triangles with maximum angle at most 150° and minimum height $\Omega(s)$, adding new Steiner points only along edge e.*

Proof. Project all vertices of the staircase, except the first and last, diagonally (that is, along a line of slope -1) onto e. This cuts M into some number of trapezoids along with two other faces, one at each end of the staircase.

Since e has positive slope, the angles where diagonals of slope -1 hit e measure less than $135°$. The angles where these diagonals leave staircase vertices measure less than $149.04°$, the sum of $135°$—the angle between a diagonal and a true horizontal or vertical—and the angle $14.04°$ by which an almost-horizontal or almost-vertical may deviate from the true direction. Hence all trapezoids have maximum angle less than $150°$.

What about the end faces? Assume for the moment that e has slope at most 1 as in Figure 4. Because we retained lines supporting occupied and neighbor squares, the (unwarped) lower left end face cannot be taller than it is wide. This means that its lower right corner can be projected diagonally onto e, and hence this end face is a convex quadrilateral. Similar reasoning proves that end faces must be convex quadrilaterals or triangles with maximum angle $150°$. Hence either diagonal triangulates a quadrilateral end face with maximum angle at most $150°$.

Height $\Omega(s)$ comes for free. All edges in faces along e have lengths $\Omega(s)$ before projecting diagonally. All slopes are bounded away from -1, so this property is preserved by the projection. □

Lemma 5 shows how to triangulate the faces along one edge of P'. We perform the same steps for each edge of $\partial P'$, treating negative-slope edges symmetrically. An edge e processed after the first edge may define more than one monotone polygon if e participates in a previous edge's monotone polygon. In other words, when we come to edge e, we merge consecutive faces along e that have not already been merged into previous edges' monotone polygons. Lemmas 4 and 5 apply to any polygon resulting from merging a connected sequence of faces along e.

We have now shown how to triangulate all faces touching edges of $\partial P'$. The lopped-off isosceles triangles can be cut into a fan triangulation with maximum angle $135°$ and minimum height $\Omega(s)$. Simply splitting interior rectangles with diagonals, we obtain an $O(n^2)$-size triangulation.

We now work on improving the size bound. We start by merging all interior rectangles as in Section 2, obtaining a set of rectilinear polygons. We obtain simple rectilinear polygons in the case that P' is simple, rectilinear polygons with holes in the case that P' has holes.

3.1. Simple Polygons

We now show how to triangulate simple rectilinear polygons without large angles. A *subdivided rectangle* is a rectangle with some number (possibly zero) of additional vertices along its four sides. A *set of matching subdivided rectangles* is a collection of subdivided rectangles, such that each rectangle satisfies the matching condition of Definition 2 (that is, each vertex has a partner opposite), and abutting rectangles share vertices (that is, each vertex must be a vertex of each rectangle it touches).

Lemma 6. *A simple n-vertex rectilinear polygon Q can be partitioned into a set of matching subdivided rectangles with a total of $O(n \log n)$ vertices.*

Proof. We start by sorting the x- and y-coordinates of all vertices of Q. We

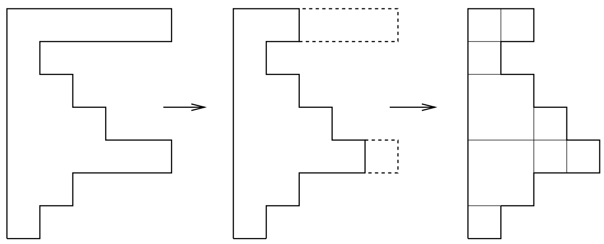

Figure 5. Dividing a rectilinear polygon into a set of matching subdivided rectangles.

then map coordinates to integers between 1 to n, preserving ties (that is, two identical coordinates must map to the same integer). We now have a polygon Q_G equivalent to Q embedded on an $n \times n$ square grid. A partition of Q_G into a set of matching subdivided rectangles induces a partition of Q with the same total number of vertices.

Next we "shrink" Q_G into another equivalent polygon Q'_G with total boundary length $O(n)$. Draw in all horizontal lines from vertices of Q_G, right or left across the interior until hitting the boundary. Now replace all vertical edges in this *horizontal visibility diagram* by ones of unit length (the first step shown in Figure 5). This may change the polygon into one that requires Riemann sheets to embed it. After shrinking all vertical edges, remove the horizontal visibility lines. Then draw the vertical visibility diagram, shrink all horizontal edges, and remove the visibility lines. Now we have polygon Q'_G with boundary length $O(n)$ (in fact, at most $2n$). A partition of Q'_G induces a partition of Q_G, and hence of Q, with the same complexity.

The final step is the actual partitioning. Our strategy is to cut out unit squares so that all corner vertices (i.e., ones with angle 90° or 270°) have coordinates congruent to 0 mod 2, then 0 mod 4, and so forth. Consider a horizontal edge e of Q'_G that has an odd y-coordinate. Removing a row of unit squares along e removes two corners with odd y-coordinates, and introduces no new odd x- or y-coordinates to corners. By removing $O(n)$ unit squares, we can ensure that all corners have even coordinates. See Figure 5. We now repeat the process, removing 2×2 squares to ensure that all corners have coordinates congruent to 0 mod 4. Notice that at each stage, the squares removed have total perimeter length $O(n)$. We do this $O(\log n)$ times, until all of Q'_G is tiled by squares. Each square has no more subdivision points than its perimeter length, so the total number is $O(n \log n)$. $\square$

Lemma 7. *A subdivided rectangle, satisfying the matching condition, with subdivision points at least b apart, can be triangulated into $O(n)$ triangles with maximum angle 135° and minimum height $\Omega(s)$, adding Steiner points only to its interior.*

Proof. Assume w.l.o.g. rectangle R has horizontal dimension h and vertical dimension v, with $h \geq v$. We add Steiner points only along a horizontal line ℓ midway between R's top and bottom sides. Consider translating a point x along ℓ, starting from R's left side. When x is distance at least $\max\{0, v/2 - b\}$ from R's left side, then any triangle with apex x and a base that is a subsegment of R's

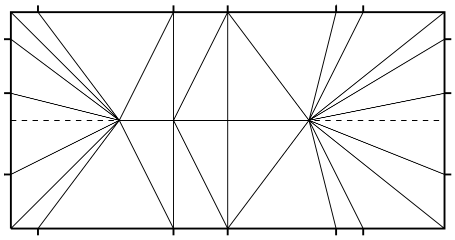

Figure 6. Triangulating a subdivided rectangle with maximum angle $135°$ and height $\Omega(s)$.

left side will be acceptable. At distance at most $v/2 + b$, any triangle with apex x and a base that is a subsegment of R's top side, to the left of x, will also be acceptable. Thus we start by placing Steiner points along ℓ at distance d_ℓ and d_r from the left and right sides of R, respectively. Distances $d_\ell, d_r > 0$ are chosen so that $v/2 - b \leq d_\ell, d_r \leq v/2 + b$ and the two Steiner points are either identical or at least b apart (and either identical or b away from each projection of a top or bottom subdivision point). We can now add each triangle with apex at the left Steiner point and base on the left side or along the top or bottom, with rightmost point at distance at most $v - b$ from the left side. We treat the right side similarly. The middle stretch of R can be triangulated using Steiner points that are projections of subdivisions from top and bottom. See Figure 6. □

Theorem 2. *In time $O(n \log n)$, an n-vertex simple polygon can be triangulated into $O(n \log n)$ triangles of height $\Omega(s)$ and maximum angle $150°$.*

Proof. We start with the corner-lopping, dicing, and warping steps of Section 2, then repair boundary faces as in Lemma 5. We merge all interior rectangles, and apply Lemmas 6 and 7 to the merged region. This results in a triangulation of size $O(n \log n)$. Achieving the running time is straightforward. □

3.2. Polygons with Holes

Lemma 6 fails for rectilinear polygons with holes. We now give a lower bound example of a rectilinear polygon with holes for which any partition into a set of matching subdivided rectangles has complexity $\Omega(n^{3/2})$. The example consists of a square of side n containing a regular square grid of $n - 4$ point (or tiny square) holes. The grid of holes has spacing about $n^{1/2}$; we rotate it slightly with respect to the square frame so that the difference between the y-coordinates of two successive holes in the same row is about 1. The minimum spanning tree of the polygon's vertices has total length $\Omega(n^{3/2})$ in the L_1 metric. Notice that any partition of the input into rectangles forms a network that spans the vertices, because no hole can be interior to a rectangle. The number of subdivisions along a rectangle side must be at least its L_1 length, since the matching condition forces each subdivision point to continue across the entire figure. Therefore the total number of subdivisions is at least the total perimeter length of the rectangles, which is at least the length of the minimum spanning tree, and hence $\Omega(n^{3/2})$.

We now show that this $\Omega(n^{3/2})$ bound is tight by giving an algorithm for partitioning polygons with holes. (An alternative approach connects holes to the boundary to form a simple polygon and then applies Lemma 6.)

Lemma 8. *An n-vertex rectilinear polygon Q with holes can be partitioned into a set of matching subdivided rectangles with a total of $O(n^{3/2})$ vertices.*

Proof. Imagine a grid (of size $O(n^2)$) formed by extending each vertex of the polygon vertically and horizontally. This grid serves as a guide as we partition the polygon.

The initial partition is a set of infinitely-tall rectangles, formed by choosing vertical grid lines spaced about $n^{1/2}$ grid lines apart. This partition divides the plane into at most $n^{1/2}$ rectangles, each with $O(n)$ subdivision points on its boundary.

We now describe how to refine the initial partition. As long as some rectangle in the partition contains a polygon vertex, we split that rectangle by passing a horizontal line through the vertex, stopping the segment at each end when it intersects a vertical chosen in the initial partition. The horizontal segment crosses about $n^{1/2}$ vertical grid lines, and hence requires $O(n^{1/2})$ subdivision points on it (projections from above and below) to satisfy the matching condition. If continued to the left and right, the horizontal segment would cross at most $n^{1/2}$ vertical rectangle boundaries, so again only $n^{1/2}$ new subdivision points are needed to satisfy the matching condition. After adding all $O(n)$ such horizontals, the total number of subdivision points is $O(n^{3/2})$, and the partition divides Q into a set of matching subdivided rectangles. □

Theorem 3. *In time $O(n \log n + k)$, an n-vertex polygon with holes can be triangulated into $k = O(n^{3/2})$ triangles of height $\Omega(s)$ and maximum angle $150°$.* □

4. Polygons with Internal Boundaries

In this section, we discuss the minmax-height and no-large-angle problems for polygons with internal boundaries; this type of input arises in simulating physical devices made of more than one material. Most generally we could allow the input to be a *planar straight-line graph*, that is, a planar embedding of a planar graph so that each edge is a line segment of finite length. (In order to triangulate a PSLG with finite-size triangles, one typically encloses it in a bounding box or triangulates only the bounded faces.) We have no new positive results for polygons with internal boundaries; instead we give two counterexamples.

First we show a lower bound example due to Paterson[18] for the no-large-angle problem. (This example also appeared in Ref. 3; we repeat it here for completeness.) Consider the case of a convex polygon with an "accordion" triangulation, as shown in Figure 7. There are $\Theta(n)$ long, skinny nearly-horizontal triangles; there are also $\Theta(n)$ vertices evenly spaced along the top of this stack of triangles. Because each vertex at the top must have a "downwards" edge, and each of the endpoints of these edges must in turn have downwards edge, there must be a path running down from each top vertex through the stack of skinny triangles. If triangles are sufficiently

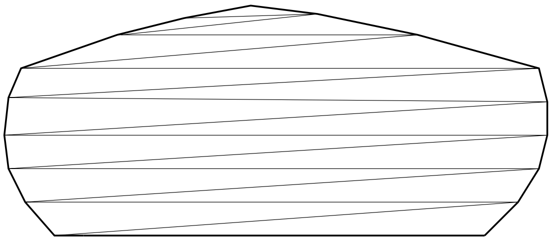

Figure 7. This PSLG requires $\Omega(n^2)$ triangles with angles bounded below 180°.

skinny, separate paths cannot merge because of the angle bound. Thus each path must have complexity $\Omega(n)$ and the entire triangulation complexity $\Omega(n^2)$.

Section 3 showed how to simultaneously achieve maximum angle 150° and minimum height a constant fraction of optimal for polygons with holes. We now show that both criteria cannot be simultaneously satisfied for planar straight-line graphs. Figure 8 shows a very skinny triangle sitting on top of a rectangle with two closely-spaced subdivision points along its base. Say the coordinates of the subdivision points are $(k, 0)$ and $(k+1, 0)$, and the coordinates of the the rectangle are $(0, 0)$, $(0, 1)$, $(k^2, 1)$ and $(k^2, 0)$. Height $\Omega(s)$ can be achieved only with a Steiner point x near the subdivision points; x cannot lie on the top side of the rectangle, as this decreases the feature size of the triangle to about $1/k$. So x must be interior to the rectangle, and then there must be a large angle (arbitrarily close to 180° as k grows) either at x or at some other interior Steiner point.

Figure 8. This PSLG forces either large angle or small height.

5. Nonobtuse Triangulation of Convex Polygons

Assume P is a region bounded by a single convex polygon. In this section, we show how to triangulate P with $O(n^{1.85})$ nonobtuse triangles. We first describe our earlier $O(n^2)$ algorithm for convex polygons[3]; our new algorithm builds on the older one. We orient the polygon so that its longest diagonal is horizontal. (We assume that the longest diagonal is not part of ∂P; otherwise our construction can be somewhat simplified.) By reflection, we may assume that the vertex of maximum y-coordinate has x-coordinate at least as large as the x-coordinate of the vertex of minimum y-coordinate.

We start the construction by drawing a vertical line segment through each vertex of P extending to the boundaries of the polygon. These lines divide the polygon into quadrilaterals with two vertical sides. Each vertex of a quadrilateral will either be an

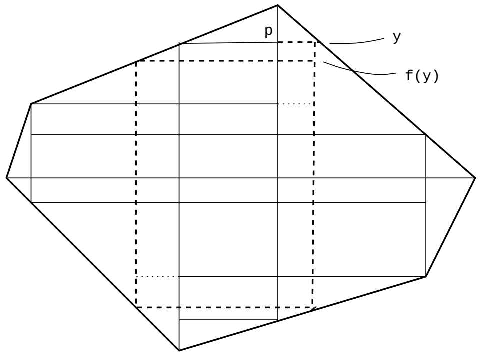

Figure 9. Convex polygon with bounced path.

original input vertex, or a point where a vertical line touches the polygon boundary. Next we draw a horizontal line segment through each vertex and extend each line segment to the last possible vertical segment. In other words, each endpoint of a horizontal segment should lie either on a vertical segment, or on the vertex inducing the horizontal, and each horizontal segment should be as long as possible with this property. Notice that the entire longest diagonal appears in the partition.

Consider a moving particle within P that travels along horizontal and vertical lines. If the particle starts by moving vertically downwards from a point along the upper right side of ∂P (that is, between the vertex of maximum y-coordinate and the vertex of maximum x-coordinate), then three bounces (moving first down, then left, then up, and finally right) return the particle to the upper right side. If the particle starts by moving downwards from the upper left side, however, then it may strike either the lower left or lower right side of ∂P first. If the particle strikes the lower left, three bounces return it to the upper left, otherwise three bounces sends it to the upper right.

We now show how to use the analogy of a bouncing particle to extend and simplify the grid partition of P. Consider a subdivision point p on a triangle on the upper right side of P as shown in Figure 9. We can bounce a path from p as follows. We first draw a horizontal segment from p extending to the right until meeting ∂P, and then a vertical where the horizontal meets the boundary. We now must lengthen (shown dotted) each other horizontal segment for which the new vertical is now the last vertical. We then draw a horizontal segment extending to the left from the lower endpoint of the new vertical, and repeat the process. Finally a last horizontal returns the path to the chain of triangles on the upper right side of P (at the point marked $f(y)$ in Figure 9).

Our goal is to bounce paths until each triangle has at most one subdivision point. When we have achieved this condition, the following fact lets us complete the nonobtuse triangulation.

Lemma 9. *If a right triangle has one subdivision point on a leg, it can be triangulated with three right triangles, by adding a diagonal from the opposite vertex to the subdivision point, and then dropping an altitude to the hypotenuse.* $\square$

We now construct an advantageous order in which to bounce subdivision points. We first bounce all subdivision points that do not return to their own side of ∂P. These are the subdivision points that first form verticals with x-coordinates between the x-coordinates of the highest and lowest vertex of P. Once we have bounced these subdivision points, all subdivision points are the sort that return to their own sides such as p in Figure 9.

For a subdivision point p at height y from the longest diagonal, define $f(y)$ to be the height of the new subdivision point that would be created by bouncing a path from p. Function $f(y)$ is continuous, monotone, and piecewise linear with $O(n)$ breakpoints in linearity. As a consequence, $f(y) - y$ is also continuous and piecewise linear.

We partition the plane into horizontal strips so that within each strip $f(y) - y$ has the same sign (positive, negative, or zero). We extend the partition by bouncing paths at the $O(n)$ heights at which $f(y) - y$ changes sign. These paths are closed and therefore do not introduce new subdivisions; they ensure that no triangle of the partition contains portions of more than one strip. Now for each y, $f(y)$ is in the same strip as y.

Now consider a triangle with more than one subdivision point in a strip where $f(y) > y$, and let p and q be the lowest two points in this triangle, with p lower than q. Bouncing the path from q creates a new subdivision point higher than q, and cuts off a triangle in which p is the lone subdivision point. We can repeat this process until all subdivision points in the strip are alone. Each extension creates a new lone subdivision point; therefore, after $O(n)$ extensions all subdivision points are alone in their triangles. In strips where $f(y) < y$, the process is similar, beginning with the highest two points that are not alone. Finally, in a strip where $f(y) = y$, bouncing a path from a subdivision point creates a rectangle. So in this case, all subdivision points can be immediately removed.

We have now described our $O(n^2)$ algorithm: partition P as above, bounce paths until each triangle along ∂P has at most one subdivision point, and then triangulate with Lemma 9. To improve to $o(n^2)$, we merge rectangles again.

5.1. Merging rectangles

The boundary of the union of rectangles interior to P is a simple polygon Q satisfying the matching condition of Section 2. Lemma 6 states that Q can be partitioned into a set of "large" matching subdivided rectangles, with a total of $O(n \log n)$ Steiner points. Within a large rectangle, there is currently a grid of small rectangles. Our method for reducing the grid within each large rectangle is based on the following observation. See Figure 10(a).

Lemma 10. *If rectangle R has a single subdivision point on a vertical side, and its width is greater than either of the distances between the subdivision and the top or bottom, then R can be triangulated into nonobtuse triangles without introducing more subdivisions.* $\square$

Let a *column* be a set of vertically adjacent small rectangles; the *width* of the

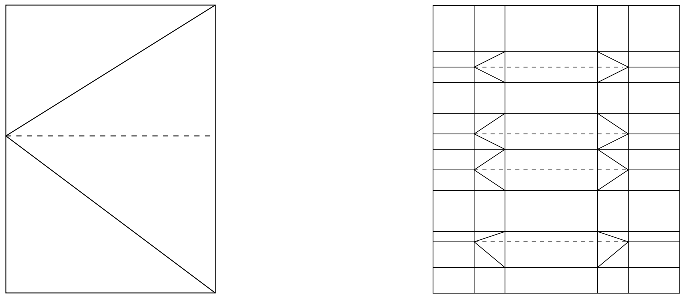

Figure 10. Removing interior lines from a subdivided rectangle: (a) Two small rectangles merge, removing a subdivision; (b) previous figure and its reverse combine to remove lines from the grid.

column is the distance between its left and right boundaries. Similarly let a *row* be a set of horizontally adjacent small rectangles, and let its *height* be the distance between top and bottom boundaries.

Assume two nonadjacent columns have widths a and b, and two adjacent rows—divided by horizontal line ℓ—have heights c and d, with $a, b > c, d$. Suppose we remove the portion of ℓ between the two columns. Lemma 10 then produces triangles at the endpoints of the removed line segment, that may not be changed by later modifications to the grid. In order to gain any benefit from this process, we must remove many horizontal lines using the same columns. Figure 10(b) illustrates several horizontal lines being removed by the same two columns. We now quantify how many lines can be removed at a time.

Lemma 11. *Let a rectangle be given with m subdivisions on its vertical sides (so it has $m+1$ rows). Let a and b be the widths of two columns, with $\min\{a, b\}$ at least as large as $(1/2+\epsilon)m$ of the row heights. Then $\epsilon m - 1$ horizontal lines can be eliminated between the columns.*

Proof. There are $(1/2-\epsilon)m+1$ rows that are too high to satisfy Lemma 10, and each one protects the lines above and below it. Thus there are $m - 2\epsilon m + 2$ lines protected, so $2\epsilon m - 2$ lines that can be removed. In the worst case, the removable are all adjacent to each other, and we can remove only half of them. $\square$

We now describe how to reduce the grid inside a large square. Suppose there are m rows and k columns. Let M_r be the 2/3-median of the row heights; i.e., $m/3$ rows are higher than M_r and $2m/3$ are less high. Similarly let M_c be the 2/3-median of the columns.

Assume $M_r \leq M_c$; the opposite case is treated symmetrically. Let l be the leftmost column with width at least M_r and let r be the rightmost such column. Then because $n/3$ rows have height at least $M_c \geq M_r$, columns l and r are separated by at least $n/3$ columns. Also note that l and r are wider than the height of $2m/3$ rows; therefore by Lemma 11 we can eliminate $m/6$ rows between the two columns.

This gives us three subproblems, which are solved recursively. The subproblems consist of (1) the columns between 1 and $l-1$, with all m rows; (2) the columns

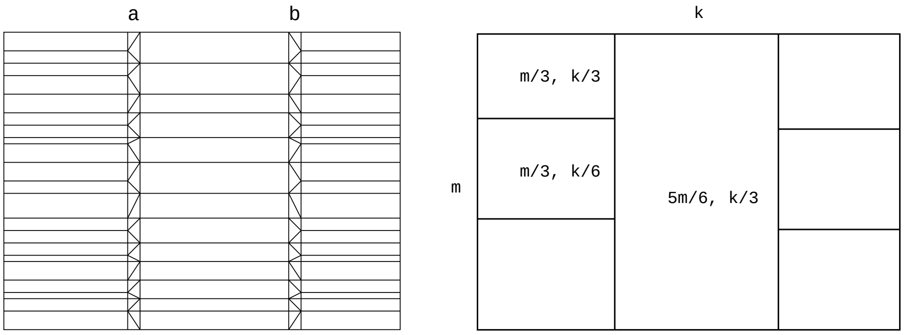

Figure 11. Recursive division into smaller rectangles: (a) single split, split into three parts with fewer rows in middle; (b) outer subproblems split again, eliminating columns.

between $l+1$ and $r-1$, with $5m/6$ rows; and (3) the columns between $r+1$ and k, with all m rows. We also triangulate columns l and r, using m triangles. This partition and row elimination are shown in figure 11(a). The total number of triangles used is

$$R(m,k) = R(l-1,m) + R(r-l-1,5m/6) + R(k-r,m) + m.$$

If we solve for the worst-case size, by allowing all possible choices of l and r (with $r-l \geq m/3$) and by including the dual case in which $M_r > M_c$, we achieve a bound that is slightly better than linear. Instead of settling for this modest improvement, we improve our algorithm with another level of recursion.

Let us examine subproblems (1) and (3). By construction, all columns in the subproblems have widths smaller than M_r, which remains the 2/3-median of the row heights in the subproblems. Therefore, by choosing the topmost and bottommost rows with heights at least M_r, we can apply Lemma 11 to eliminate half of the columns between the two rows. Unfortunately this second level of elimination cannot be continued to a third level. We first split the problem according to *columns* that were chosen to be larger than the 2/3-median of the *rows*; in the second level we chose the *rows*, but compared them to the *row* median. So the second level recursion is not the same as the first level, and the trick cannot be extended.

Thus we can subdivide each of the two subproblems into three sub-subproblems, giving a recursion that divides each rectangle into seven parts, as depicted in Figure 11(b). The recurrence describing the possible arrangements of the seven parts is quite complicated. But as indicated by the numbers in the figure, we can make some simplifying assumptions about the dimensions of each subproblem. We do not attempt to prove these assumptions; instead we use them heuristically to simplify the recurrence, giving a solution as a function of m and k. We later confirm the solution. Our assumptions are as follows:

- In the worst case, subproblems (1) and (3) have equal numbers of columns, $l-1 = k-r$. Similarly, the corresponding sub-subproblems have equal numbers of rows.

- In the worst case, subproblem (2) has exactly $m/3$ columns, and the corresponding sub-subproblems have exactly $k/3$ rows.
- If $m < k$ the worst case happens when $M_r > M_c$ and the first level recursion splits the problem along rows of the rectangle. If $m > k$ the worst case happens when $M_r < M_c$ and the rectangle splits by columns.

With these assumptions, we can describe the number of triangles used by the following recurrence, for the case $m < k$.

$$R(m,k) = R(m/3, 5k/6) + 2R(m/6, k/3) \\ + 4R(m/3, k/3) + O(k).$$

The first term represents subproblem (2), the second term represents the two corresponding sub-subproblems, and the third term represents the four remaining sub-subproblems.

The recurrence solves to $O(m^{0.8475}k)$. The exponent in the solution was derived numerically as an approximate solution to the equation

$$(5/6 + 4 \cdot 1/3)(1/3)^x + (2 \cdot 1/3)(1/6)^x = 1.$$

We show below that our heuristic assumptions are justified, and that this bound actually holds for our algorithm. Let us summarize our results.

Lemma 12. *A rectangle subdivided into m rows and k columns, with $m < k$, can be triangulated using $O(m^{0.8475}k)$ nonobtuse triangles, without introducing any new subdivisions.* □

Theorem 4. *In time $O(n^{1.848})$, an n-vertex convex polygon can be triangulated with $O(n^{1.848})$ nonobtuse triangles.*

Proof. First consider running the $O(n^2)$-algorithm. Then using Lemma 6, we merge interior rectangles into a set of matching subdivided rectangles with a total $O(n \log n)$ subdivision points. By Lemma 12, we can triangulate these rectangles with $O((n \log n)^{1.8475})$ triangles, adding only interior Steiner points.

To improve the running time to $O(n^{1.848})$, we form the rectilinearly convex interior Q, bounce paths, and divide Q into large rectangles, all in time $O(n \log n)$. Within each large rectangle, we represent the grid implicitly in terms of row heights and column widths. The computation time will then be governed by the same recurrence as the number of triangles. □

5.2. Complete solution of recurrence

We now prove that the true recurrence describing our rectangle triangulation algorithm has the solution we claimed. We assume throughout that $m < k$.

First suppose that the initial subdivision is by rows. Then the following recurrence describes the number of triangles needed. Here $m_1 + m_2 + m_3 = m$, $k_1 + k_2 + k_3 = k_4 + k_5 + k_6 = k$, $m_2 \geq m/3$, $k_2 \geq k/3$, and $k_5 \geq k/3$. Note that $m' < k'$ for all subproblems involved, so the inductive hypothesis can be used directly.

$$\begin{aligned}
R(m,k) &= R(m_1,k_1) + R(m_1/2,k_2) + R(m_1,k_3) \\
&\quad + R(m_2, 5k/6) \\
&\quad + R(m_3,k_4) + R(m_3/2,k_5) + R(m_3,k_6) \\
&\quad + O(k) \\
&= cm_1^{0.8475}k_1 + c(m_1/2)^{0.8475}k_2 + cm_1^{0.8475}k_3 \\
&\quad + cm_2^{0.8475}(5k/6) \\
&\quad + cm_3^{0.8475}k_4 + c(m_3/2)^{0.8475}k_5 + cm_3^{0.8475}k_6 \\
&\quad + O(k) \\
&\le cm_1^{0.8475}(0.85192k) + cm_2^{0.8475}(0.83334k) \\
&\quad + cm_3^{0.8475}(0.85192k) + O(k) \\
&= c(m_1^{0.8475} + m_2^{0.8475} + m_3^{0.8475})(0.85192k) \\
&\quad - cm_2^{0.8475}(0.01858k) + O(k) \\
&\le c(1.1824m^{0.8475})(0.85192k) \\
&\quad - c(0.39414m^{0.8475})(0.01858k) + O(k) \\
&\le 0.99999cm^{0.8475}k + O(k).
\end{aligned}$$

So if c is large enough the result is $O(m^{0.8475}k)$ as claimed.

In the other case, the initial subdivision is by columns. For any possible such subdivision, we can construct an alternate subdivision by rows, in which the ratios of k_i/k and m_i/m switch roles to match the recurrence above. So in the alternate problems, the products m_ik_i are the same as they would be in the original problems. But the *aspect ratio* k_i/m_i is larger in the alternate subproblems than in the originals. In some cases the original aspect ratio may even be smaller than one (i.e., in the original problem, k_i may be smaller than m_i). But in the latter case m_i/k_i will be smaller than the alternate value of k_i/m_i. In the inductive hypothesis, reducing the aspect ratio and fixing the product $m'k'$ can only decrease the total number of triangles. So subdivision by columns is preferable to subdivision by rows, and cannot occur as the worst case.

6. Conclusions

We have given a variety of guaranteed-quality triangulation methods and devised three different techniques to avoid the quadratic blow-up of rectilinear grids: the sweep method that omits every other point, the partitioning of rectilinear polygons into subdivided rectangles, and finally the simultaneous horizontal-vertical divide-and-conquer of Section 5. There are a number of open problems.

- **Maxmin height for planar straight-line graphs.** Does there always exist a Steiner triangulation with height $\Omega(s)$? Examples related to Figure 8 suggest not.

- **Better angle bounds.** For the sake of simplicity, we did not try to improve our angle bound beyond 150°. Is there a reasonably simple algorithm that guarantees a better angle bound? Is it possible to achieve all angles strictly smaller than 90° (some by tiny amounts dependent upon the geometry) with a polynomial number of triangles?
- **Lower bounds.** Does it require a superlinear number of Steiner points to triangulate a polygon so that all angles are smaller than, say, 150°? Does it require $\Omega(n^{3/2})$ for a polygon with holes?
- **Interior Steiner points.** Throughout this paper, the following subproblem occurs: triangulate a subdivided rectangle with guaranteed-quality triangles, adding only interior Steiner points. Can we characterize polygons that can be so triangulated?
- **Inside-outside nonobtuse triangulation.** Can we triangulate both the interior and exterior of a polygon (say, the region between the polygon and its convex hull) with a polynomial number of nonobtuse triangles? Solving this problem would also give another solution to the problem of finding a "conforming Delaunay triangulation".[19]

Acknowledgements

We would like to thank the referees for some helpful suggestions and Scott Mitchell and Mike Paterson for some lower bound examples.

References

1. M. Bern and D. Eppstein. Mesh generation and optimal triangulation, *Computing in Euclidean Geometry*, D.-Z. Du and F.K. Hwang, eds., World Scientific, 1992.
2. M. Bern, D. Eppstein, and J.R. Gilbert. Provably good mesh generation, *31st IEEE Foundations of Computer Science*, 1990, 231–241.
3. M. Bern and D. Eppstein. Polynomial-size nonobtuse triangulation of polygons, *7th ACM Symp. on Computational Geometry*, 1991, 342–350.
4. B. Baker, E. Grosse, and C. Rafferty. Nonobtuse triangulation of polygons, *Discrete and Comp. Geom.* 3 (1988), 147–168.
5. E. Melissaratos and D. Souvaine. Coping with inconsistencies: a new approach to produce quality triangulations of polygonal domains with holes, *8th ACM Symp. on Computational Geometry*, 1992, 202–211.
6. D. Eppstein. Approximating the minimum weight triangulation, *3rd ACM-SIAM Symp. on Disc. Algorithms*, 1992, 48–57.
7. S. Mitchell and S. Vavasis. Quality mesh generation in three dimensions, *8th ACM Symp. on Computational Geometry*, 1992, 212–221.
8. R. Barnhill. Computer aided surface representation and design, *Surfaces in CAGD*, R. Barnhill and W. Boehm, eds., North-Holland, 1983, 1–24.
9. C. Gold, T. Charters, and J. Ramsden. Automated contour mapping using triangular element data structures and an interpolant over each irregular triangular domain, *Proc. Siggraph*, 1977, 170–175.

10. J. Gregory. Error bounds for linear interpolation on triangles, *The Mathematics of Finite Elements and Application II*, J. R. Whiteman, ed., Academic Press, 1975, 163–170.
11. I. Babuška and A. Aziz. On the angle condition in the finite element method, *SIAM J. Numer. Anal.* 13 (1976), 214–227.
12. S. Salzberg, A. Delcher, D. Heath, and S. Kasif. Learning with a helpful teacher. *12th Int. Joint Conf. Artificial Intelligence*, 1991.
13. S. Mitchell. Refining a triangulation of a planar straight-line graph to eliminate large angles. *34th IEEE Foundations of Computer Science*, 1993, 583–591.
14. M. Bern, D. Dobkin, and D. Eppstein. Triangulating polygons without large angles. *8th ACM Symp. on Computational Geometry*, 1992, 222–231.
15. S. Mitchell, personal communication, 1991.
16. S. Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica* 2 (1987), 153–174.
17. A. Lingas. Voronoi diagrams with barriers and their applications. *Inform. Process. Letters* 32 (1989), 191–198.
18. M.S. Paterson. Personal communication, 1990.
19. H. Edelsbrunner and T.S. Tan. An upper bound for conforming Delaunay triangulations, *8th ACM Symp. on Computational Geometry*, 1992, 53–62.