Provably Good Mesh Generation

Marshall Bern* David Eppstein*† John Gilbert*

Abstract

We study several versions of the problem of generating triangular meshes for finite element methods. We show how to triangulate a planar point set or polygonally bounded domain with triangles of bounded aspect ratio; how to triangulate a planar point set with triangles having no obtuse angles; how to triangulate a point set in arbitrary dimension with simplices of bounded aspect ratio; and how to produce a linear-size Delaunay triangulation of a multi-dimensional point set by adding a linear number of extra points. All our triangulations have size (number of triangles) within a constant factor of optimal, and run in optimal time $O(n \log n + k)$ with input of size n and output of size k. No previous work on mesh generation simultaneously guarantees well-shaped elements and small total size.

1. Introduction

Geometric partitioning problems ask for the decomposition of a geometric input into simpler objects. These problems are fundamental in many areas, such as solid modeling, computer-aided design, graphical rendering, and scientific computation. Some geometric decompositions are binary space partitions, epsilon nets, convex decomposition, triangulations and tetrahedralizations, and k-D trees, quadtrees, and their relatives.

A partitioning problem of particular interest in computational geometry is optimal triangulation of a planar point set [6]. This problem finds application in cartography, spatial data analysis, and finite element methods. Optimization criteria include maximizing the minimum angle (solved by the well-known Delaunay triangulation [24, 27]), minimizing the maximum angle [13], minimizing a maximum min-containment ellipse [11], and minimizing total length (an outstanding open problem in the field [16, 20]). Variants of these problems allow one to add extra vertices, called *Steiner points*, in order to further improve the quality of the solution.

In this paper we use quadtrees to solve several "Steiner triangulation" problems motivated by finite element methods. A point set or polygon is to be triangulated, with Steiner points allowed, into "well-shaped" triangles. Though the literature contains extensive work on mesh generation algorithms (some using quadtrees), this paper is the first to simultaneously optimize element shape and total number of triangles. Some of our results generalize to higher dimensions, for which we know of no previous guarantees on either measure.

*Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304.

†Department of Information and Computer Science, Univ. of California, Irvine, CA 92717.

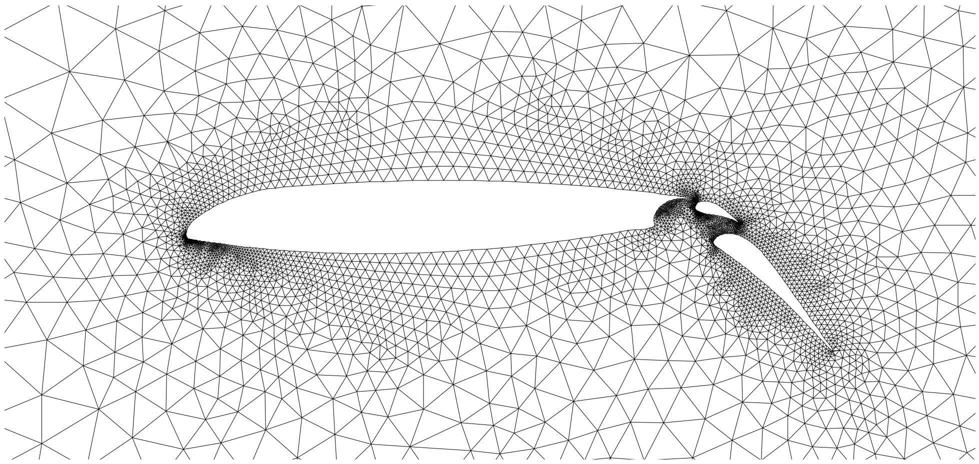

Figure 1. Part of a triangulation of a region with three holes (Barth and Jespersen).

1.1. Motivation

The finite element method [29] is a collection of techniques for approximating continuous problems by finite structures. The domain is subdivided into a *mesh* of polygonal or polyhedral *elements*, and the function of interest is approximated by a piecewise polynomial on the elements. We consider the most common case, in which the domain is a subset of the plane or of $\mathcal{R}^d$, and the elements are triangles or simplices. The mesh must satisfy several conditions, depending on the problem.

- The mesh must conform to the boundary of the region, which may consist of more than one connected component (e.g., in Figure 1 the boundary includes the three airfoils).
- The mesh must be fine enough to produce an acceptable approximation to the original problem. Parts of the domain where the solution is complicated or rapidly changing may require much smaller elements than other parts.
- The number of elements in the mesh should be small, because the complexity of solving the finite element problem depends on the mesh size.
- The individual elements must be "well-shaped". There are two important restrictions:

  No small angles. For some methods, elements with small angles lead to ill-conditioned linear systems that are difficult to solve accurately [15]. Angles close to 180° present further problems [1].

  No obtuse angles. In two dimensions, some methods require the center of the circumcircle of each element to lie in the element [2, 4], so that the perpendicular bisectors of element edges form the planar dual of the mesh. Circumcenters lie within their (closed) elements if and only if no angle is greater than 90°. For circumcenters to be well separated from the element boundary, all angles should be bounded away from 90°.

1.2. Summary of results

We consider the following problems, obtaining the results described.

1. *2D point set triangulation with no small angles.* Given n points in the plane, find a triangulation (of a convex region of the plane) that includes the given points as vertices and has all angles larger than some constant (or, equivalently, the aspect ratios of all triangles smaller than some constant). We give an algorithm to produce such a triangulation of size within a constant factor of the minimum possible size. The size of the triangulation is also bounded by $O(n \log A)$, where A is the worst aspect ratio in a Delaunay triangulation of the original point set. In addition, the triangulation can be constructed to have no obtuse angles at the cost of a larger constant factor in size.
2. *2D point set triangulation with no obtuse angles.* Given n points in the plane, find a triangulation with no obtuse angles. We give an algorithm to produce such a triangulation of size $O(n)$. Thus for some point sets, forbidding small angles requires a much larger triangulation than forbidding obtuse angles.
3. *2D point set triangulation with only acute angles.* Given n points in the plane, find a triangulation with only acute angles. We modify Algorithm 1 above so that all angles are between $36°$ and $80°$. We modify Algorithm 2 so that all angles are strictly smaller than (though perhaps arbitrarily close to) $90°$.
4. *2D polygon triangulation with no small angles.* The input is a connected planar region bounded by a union of disjoint polygons (that may degenerate to paths or points); there is a lower bound on boundary angles facing the interior of the region. The problem is to triangulate the region so that each vertex of the boundary is a vertex of the triangulation, each edge of the boundary is a union of edges of the triangulation, and each angle is larger than some constant. We give an algorithm to produce such a triangulation of size within a constant factor of minimum. The angle bound we achieve is $18.4°$; that is, each new angle in the triangulation measures at least this much. The maximum angle is smaller than $153.2°$.
5. *Point set triangulation with no small solid angles.* Given n points in $\mathcal{R}^d$, find a triangulation with d-dimensional solid angles larger than some constant. Equivalently, all simplices must have bounded aspect ratio. We give an algorithm to produce such a triangulation of size within a constant factor of minimum.
6. *Linear-size Delaunay triangulation in $d \geq 3$ dimensions.* The Delaunay triangulation may have size $\Theta(n^{\lceil d/2 \rceil})$ [12]. We give an algorithm that adds $O(n)$ new points such that the Delaunay triangulation of the entire set has size $O(n)$, and, in fact, bounded vertex degree.

In addition, our methods can be easily adapted to satisfy user-supplied conditions on the degree of refinement in various areas. All our algorithms run in time $O(n \log n + k)$, where n is the input size and k is the output size. If the input includes the sorted ordering in each coordinate, the running times of all except Algorithm 4 are $O(n + k)$.

1.3. Related work

Mesh generation has been the subject of a great deal of work, both practical and theoretical. However, very little previous work offers guarantees, and none offers simultaneous guarantees on mesh quality and size.

Thacker [30] and Shephard [26] survey the extensive literature of heuristics. Bank [3], Joe [17], Yerry and Shephard [31] (who use quadtrees), and many others [6] have written automatic mesh generation programs, but the outputs of these programs have no proven quality or size bounds.

On the theoretical side, Baker et al. [2] give an algorithm to triangulate the interior of a simple polygon with elements whose angles are between 13° and 90°. The number of triangles used by their algorithm may be unnecessarily large; however, they suggest that quadtrees might improve the size of the triangulation. Our paper follows up on this suggestion, as well as giving an innovative size bound.

Smith [28] shows how to triangulate a polygon with elements of bounded aspect ratio but with no size bound in general. Chew [10] shows how to triangulate suitable polygonally bounded regions with approximately equal-sized elements having no angle less than 30°. Subject to a restriction on element size, the number of elements is immediately within a constant factor of optimal. Our method gives no angles less than 18.4°, but can generate meshes with elements of widely differing scales, and thus achieve optimal mesh size without restriction.

In three dimensions, Chazelle et al. [7] give an algorithm that adds $O(n^{1/2}\log^3 n)$ points and guarantees a Delaunay triangulation of size $O(n^{3/2}\log^3 n)$. Our Algorithm 6 adds more points, but achieves much smaller size. No method is known for bounded-aspect-ratio triangulation of polyhedra; triangulations with unbounded aspect ratio are known [8, 28]. Finally, the aspect ratio bound for the d-dimensional meshes generated by our Algorithm 5 implies that their skeletons have $O(n^{1-1/d})$-separators [22]. Such separators lead to efficient algorithms for a variety of problems; most relevantly, nested dissection [19] saves a factor of $n^{3/d}$ in the time to solve the linear equations that arise in the finite element method.

2. Bounded aspect ratio for point sets

The *aspect ratio* of a convex body is the ratio between its longest dimension and its shortest dimension. For a triangle abc, the aspect ratio $A(a,b,c)$ is the length of the hypotenuse (longest side) divided by the length of the altitude from the hypotenuse. The aspect ratio of a triangle is closely related to its sharpest angle θ, since $|1/\sin\theta| \le A(a,b,c) \le |2/\sin\theta|$. Another natural measure of sharpness is the ratio $R(a,b,c)$ between a triangle's longest and shortest sides. $A(a,b,c) > R(a,b,c)$ but $R(a,b,c)$ may be much smaller than $A(a,b,c)$. We write $|\mathcal{T}|$ for the number of vertices in triangulation $\mathcal{T}$, and $A(\mathcal{T})$ for the maximum value of $A(a,b,c)$ over all triangles abc in $\mathcal{T}$. Similarly $R(\mathcal{T})$ is the maximum of $R(a,b,c)$.

The main result of this section is Theorem 1 below. This theorem claims that, given a planar point set X, we can compute a triangulation with vertex set including X, with triangles of aspect ratio at most 4, and with total number of triangles at most a constant times the number in any triangulation including X that has aspect ratio at most 4. An algorithm based on quadtrees proves Theorem 1. This algorithm is our most basic result;

the remainder of the paper comprises numerous variations.

Our algorithm uses a *quadtree*, a geometrical division of the plane into a tree of square *boxes* [25]. Each box is either a leaf of the tree, or is *split* into four equal-area children. A box has four possible *neighbors* in the four cardinal directions; a neighbor is a box of the same size sharing a side. A *corner* of a box is one of the four vertices of its square. The corners of the quadtree are the points that are corners of its boxes. We say that the side of a box is *split* if either of the neighboring boxes sharing it is split. All our quadtrees are *balanced*: any side of an unsplit box may contain only one quadtree corner in its interior.

We now show how to produce the quadtree triangulation $\mathcal{QT}(X)$ for an input point set X. We normally start with a root box twice as large as, and concentric with, a minimum bounding square of X. Below we vary the choice of root box, for the purpose of proof only. All we really require of the root box is that its side length is at most a constant times the diameter of X, and that its sides lie sufficiently far from all points X.

An *extended neighbor* of a box b is another box the same size sharing either a side or a corner of b. Box b is *crowded* if b contains at least one point of X, and one or more of the following conditions holds.

C1. Box b contains two points of X.

C2. Box b has side length ℓ, and contains a single point x with a nearest neighbor in X closer than $2\sqrt{2}\ell$ units away.

C3. One of the extended neighbors of b is split.

While there is any crowded box b, we split b, and if necessary split b's extended neighbors so b's children have all eight extended neighbors. We also split any boxes necessary to maintain the balance property. After all splitting has been done, every leaf box containing a point of X is surrounded by eight leaf boxes of the same size.

Then we "warp" the quadtree framework as follows. Let y be the corner nearest x of the box containing x; we replace y by x as a corner of the quadtree. Finally, we triangulate the resulting planar subdivision. Unwarped boxes are triangulated with isosceles right triangles by adding a point in the center. Only boxes with unsplit sides have warped corners; for these we choose the diagonal that gives better aspect ratio. Figure 2(a) shows a triangulation resulting from a slightly different version of this method. Figure 2(b) shows the triangulation after some simple heuristics have reduced its size while preserving the aspect ratio bound.

Lemma 1. *The method above gives triangulations $\mathcal{QT}(X)$ with $A(\mathcal{QT}(X)) \leq 4$.*

Proof: The isosceles right triangles used to triangulate the unwarped boxes have aspect ratio 2. If a box—with side length ℓ—is warped, we have two cases.

In the first case, the input point is inside the square of the original box. Then we assume that the diagonal touching the warped point is chosen; otherwise the aspect ratio can only be better than what we prove. Consider one of the two triangles formed, with corners the input point and two other box corners. The maximum length hypotenuse is formed when the warped point is on its original location, and has length $h = \ell\sqrt{2}$. The minimum area is formed when the point is in the center of the square, and has area $a = \ell^2/4$. The maximum aspect ratio is therefore at most $h^2/2a = 4$.

In the second case, the input point is outside the original square. Then we assume that the diagonal not touching the warped point is chosen. This divides the box into an

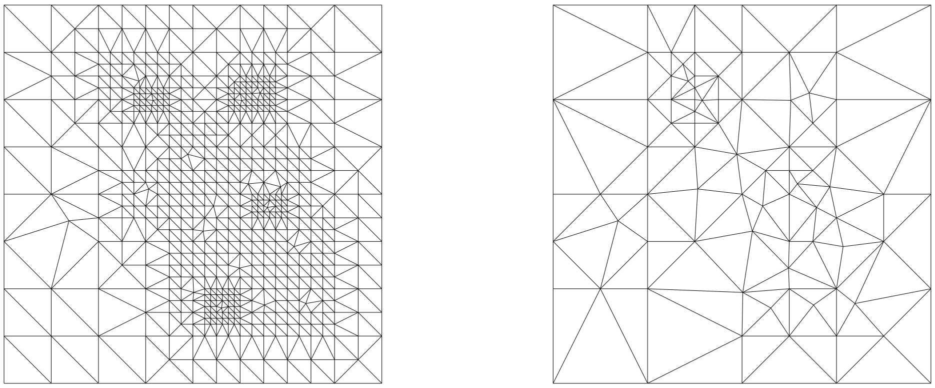

Figure 2. Triangulation of 18 random points: (a) $\mathcal{QT}(X)$; (b) heuristic size reduction.

isosceles right triangle and another triangle. If the chosen diagonal is the hypotenuse of the other triangle, then as before the area is at least $\ell^2/4$ and the aspect ratio is at most 4. Otherwise, the hypotenuse touches the input point. The altitude is minimized when the triangle is isosceles with as sharp an angle as possible; the altitude in this case has length $(\sqrt{7}-1)\ell/\sqrt{2}$. The maximum possible hypotenuse length is $\sqrt{18}\ell$. Therefore the maximum aspect ratio is at most the quotient of the two, which is less than 3.65. ∎

An improved analysis of this algorithm would give a tighter aspect ratio bound. We omit this, as we later describe algorithms that dramatically improve the aspect ratio.

Lemma 2. *There is a constant c', independent of X, such that $|\,\mathcal{QT}(X)| \le c' \cdot \sum \log R(a,b,c)$, where the sum is over all triangles in $\mathcal{DT}(X)$.*

Proof: Boxes that split to maintain the balance condition can be amortized against crowded boxes. Therefore we need only count the total number of crowded boxes in the quadtree data structure.

Linearly many crowded boxes have more than one child with points in them. It can happen at most linearly many times that a point within $2\sqrt{2}\ell$ of another point, where ℓ is the side length of the box containing the first point, becomes further away due to the shrinking sizes of boxes as they split. If a box b containing a point is split because an extended neighbor was split, but no extended neighbor contains any points, then, when either b or b's parent was split, a nearby point became farther away than $2\sqrt{2}\ell$. Again, this can only happen linearly many times.

Finally a box may contain two points, or several extended neighbor boxes may contain points, and this situation may persist when the boxes split. If splitting the children of the box or of its neighbors separates the points, we can charge linear total work. Otherwise, let Y be a maximal set of points in the union of box b and its neighbors, such that splitting b, its neighbors, or the children of b and its neighbors does not further divide Y. Then some triangle of $\mathcal{DT}(X)$ connects two points y_1 and y_2 in Y with a point z outside Y.

Each split not yet accounted for occurs between the step when Y is separated from z, and the step when y_1 and y_2 become more than $2\sqrt{2}\ell$ units apart. These steps are at most $O(\log R(y_1, y_2, z))$ quadtree levels apart, so we can charge all the crowded boxes caused by Y to triangle y_1y_2z. This triangle will not be charged by any other boxes, because once we perform the splits charged to it all three points become far away from each other in the quadtree.

Therefore the number of crowded boxes can be counted as a linear term, plus terms of the form $O(\log R(a,b,c))$ for some Delaunay triangles abc. ∎

We are now in a position to state the main result of this section.

Theorem 1. *Given any point set X, we can find a triangulation $\mathcal{QT}(X)$ such that each point of X is a vertex of $\mathcal{QT}(X)$ and $A(\mathcal{QT}(X)) \leq 4$. There is a constant c'', independent of X, such that if $\mathcal{T}$ is any triangulation containing the points of X as vertices, $|\,\mathcal{QT}(X)| \leq c'' \cdot |\mathcal{T}| \log A(\mathcal{T})$.*

Proof: Let Y be the set of vertices of $\mathcal{T}$. Lemma 2 states that there is a constant c such that $|\,\mathcal{QT}(Y)| \leq c' \cdot \sum \log R(a,b,c)$, where the triangles abc range over all triangles in the Delaunay triangulation $\mathcal{DT}(Y)$. If $Y = X$, then using the maxmin-angle characterization of the Delaunay triangulation, $A(\mathcal{T}) \geq \frac{1}{2} \cdot A(\mathcal{DT}(X)) \geq \frac{1}{2} \cdot R(\mathcal{DT}(X))$. Hence $|\,\mathcal{QT}(X)|| \leq c' \cdot \sum \log R(\mathcal{DT}(X)) \leq c' \cdot |\mathcal{T}| \log R(\mathcal{DT}(X)) \leq 2c' \cdot |\mathcal{T}| \log A(\mathcal{T})$ as required.

Otherwise, $Y \supset X$. Imagine running our algorithm on point set Y, choosing the root box for $\mathcal{QT}(Y)$ so that some subdivision of it coincides with the root box of $\mathcal{QT}(X)$. This choice of root box does not affect the lemmas above. It now follows from our construction that $|\,\mathcal{QT}(X)| \leq |\,\mathcal{QT}(Y)|$, which, by the same argument as above, is at most $2c' \cdot |\mathcal{T}| \log A(\mathcal{T}))$. Again the theorem follows. ∎

In the next section we reduce our aspect ratio bound from 4 to 5/3 at a constant factor cost in the size of the generated triangulations. Corollary 1 shows that any algorithm with a weaker aspect ratio bound can achieve at most a constant factor improvement in size. In this sense, our results are independent of our actual aspect ratio bounds.

Corollary 1. *For any $\alpha \geq 4$, let $OPT_\alpha(X)$ be the minimum size of a triangulation of X achieving aspect ratio α. Then there is a constant c_α such that $|\,\mathcal{QT}(X)| \leq c_\alpha \cdot OPT_\alpha(X)$.*

Proof: Let $\mathcal{T}$ be the triangulation achieving $OPT_\alpha(X)$. Then $|\,\mathcal{QT}(X)|$ is $O(|\mathcal{T}| \log \alpha)$, which is $O(OPT_\alpha(X))$ since α is a constant. ∎

Corollary 2. *$|\,\mathcal{QT}(X)|$ is $O(n \log A(\mathcal{DT}(X)))$.* ∎

Corollary 2 is tight, as some point sets require size $\Omega(n \log A(\mathcal{DT}(X)))$ to achieve any constant aspect ratio. An example is the set of points $(0, k\alpha)$ and $(1, k\alpha)$ for $\alpha > 1$ and $k = 1, 2, \ldots, n/2$; the aspect ratio of the Delaunay triangulation of these points is approximately α, and $\Omega(\log \alpha)$ new points must be added between successive pairs of points to interpolate between the distance within a pair and the distance between pairs. Consider a triangulation of these points that achieves, say, aspect ratio 4. There must be a triangle with an edge of length no greater than one incident to $(0, \alpha)$. Now in any "path" of triangles (i.e., a sequence of triangles such that each triangle shares an edge with its predecessor) from $(0, \alpha)$ to $(0, 2\alpha)$, the maximum possible edge length quadruples at each step. Thus such a path of triangles must have length $\log_4 \alpha$.

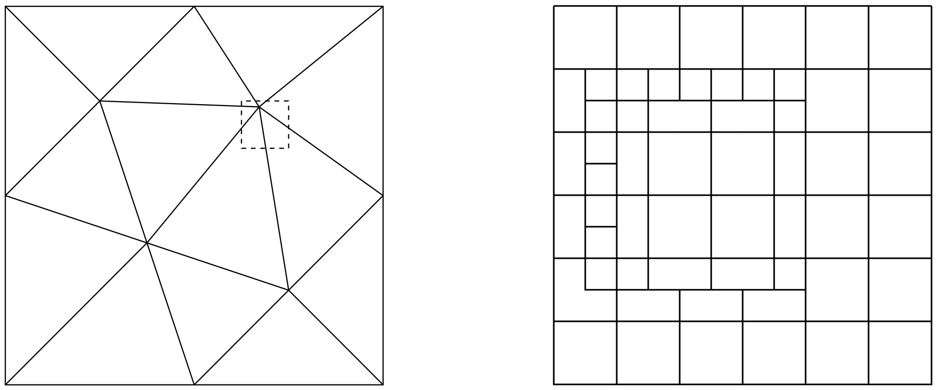

Figure 3. Non-obtuse triangulation: (a) when point is central; (b) shifted grid.

3. No obtuse angles

In this section we show how to triangulate a set of input points so that no angle is obtuse. Any triangulation without obtuse angles is a Delaunay triangulation of its vertices.

3.1. Bounded aspect ratio nonobtuse triangulation

We now describe a modification to $\mathcal{QT}(X)$ that eliminates obtuse angles while maintaining the aspect ratio bound. We first describe a solution that works when the input points are not too near quadtree box sides. Recall that, after all crowded boxes are split, the nearest quadtree corner to each input point is a corner of four equal-size *surrounding* boxes. Any box is a surrounding box of at most one point. In Figure 3(a), the large box is the union of the four surrounding boxes of an input point lying in the small dashed square. We say that point x is *central* to square s if x is contained in the square concentric with s but with half the side length. Thus each input point is central to the square that is the union of its four surrounding boxes. For now we assume that each point is also central to the box containing it. Up to rotations and reflections, the small dashed square in Figure 3(a) contains all locations central to both the containing square and the surrounding-box square.

We now add points at the centers of the boxes orthogonal to the input point. We add a point to the box diagonal from the input point, halfway between its center and the center of the square formed by the four boxes. Each surrounding box now contains one point; we connect these points to their corresponding outside corners, and also to the points in the two orthogonally adjacent surrounding boxes. Finally, we connect the input point to the point in the surrounding box diagonal from it. This construction is depicted in Figure 3(a).

Lemma 3. *The construction above triangulates the boxes surrounding an input point with no obtuse angles, and with maximum aspect ratio 2.*

Proof: It is not difficult to see that all triangles are obtuse; this also follows immediately from the aspect ratio bound. Of the 14 triangles in the figure, 8 are fixed by the construction and have maximum aspect ratio 2. The remaining six fall into three cases. We denote the length of the surrounding box sides by ℓ.

There are two triangles defined by the input point and an outside edge of the surrounding box. If the outside edge is the hypotenuse, the altitude is at least $\ell/2$ and the aspect ratio is at most 2. Otherwise, the hypotenuse length is at most $\sqrt{9/8}\ell$, and the altitude is at least $\sqrt{7/16}\ell$, so the aspect ratio is at most $\sqrt{18/7}$.

There are two triangles formed by the input point, a point in an adjacent box, and the point in the opposite box. The hypotenuse has length at most $\sqrt{9/8}\ell$, and the altitude is at least $\sqrt{81/208}\ell$, so the aspect ratio is at most $\sqrt{26/9}$.

Finally, there are two triangles formed by the input point, a point in an adjacent box, and the outside corner shared by the two boxes. If the hypotenuse's angle is fixed, the aspect ratio is maximized when the input point is on an edge of the small dashed square. If it is on the edge opposite the adjacent box, at distance $(x + 1/2)\ell$ from the outside edge of the box, then the aspect ratio can be computed to be $2(x^2 + 1)/(x + 1)$, which for $0 \leq x \leq 1/2$ is at most 2. If the point is on the other edge of the square, the altitude is minimized and hypotenuse maximized at the corner of the square, for which the aspect ratio is 17/10. ∎

Now we show how to make all points central to their surrounding boxes. Our strategy is to "shift" the grid of the quadtree near the point. Initially, each point is in the center of a three by three grid of boxes, each of size ℓ. We split each of these nine boxes, splitting other nearby boxes if necessary to maintain the quadtree balance condition. This increases the size of the construction by at most a constant factor. Our point will now be contained in the center box of a five by five grid of identically sized boxes. We split the inner nine boxes of this grid again, into boxes of side length $\ell/4$. The outer sixteen boxes are also subdivided into triangles and squares, such that their outside edges remain undivided and the input point is in the center box of a seven by seven grid. There are four possible ways of recombining the squares of this grid into a larger grid with side length $\ell/2$. In one of those ways, the input point will be central to its square. Find four such squares surrounding the grid corner nearest the input point. Remove the box sides and corners dividing those squares, which (because the grid is seven by seven) will not remove any points on the outside boundaries of the original nine boxes. The removed sides are shown as dotted lines in Figure 3(b). The remainder of the quadtree can be triangulated with isosceles right triangles.

This gives us a set of four surrounding squares for which the construction of Lemma 3 is possible. We summarize our results so far:

Theorem 2. *For any point set X, there is a triangulation containing the points of X as vertices, with no obtuse angle, with aspect ratio at most 2, and with size $O(|\mathcal{QT}(X)|)$.* ∎

3.2. Linear-size nonobtuse triangulation

If we eliminate the aspect ratio bound, similar techniques yield triangulations with linearly many new points. The only nonlinear behavior of the previous algorithm occurs when a crowded box is split without separating any input points. If this happens repeatedly, some tightly spaced cluster of points must be escaping separation by the quadtree sides. We need to "shortcut" the quadtree construction to produce small boxes around the cluster without passing through many intermediate sizes of boxes.

We triangulate the cluster recursively, resulting in a small triangulated square, which we treat as an individual point. We shift the grid so that the square is appropriately placed

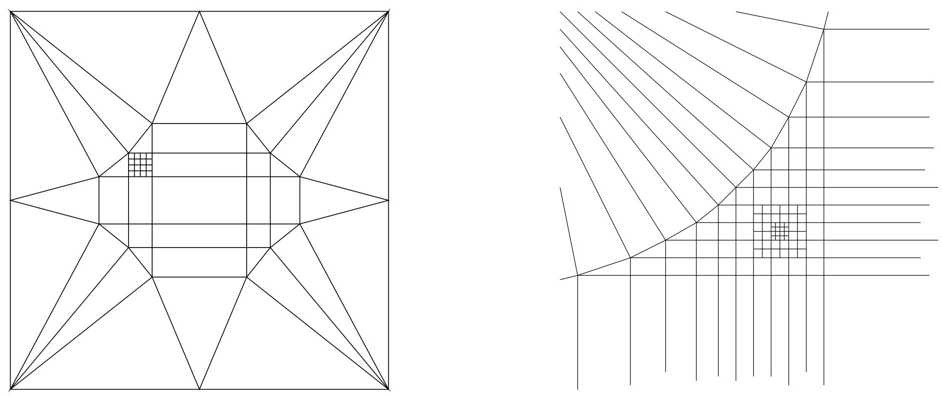

Figure 4. Connection between cluster and containing quadtree: (a) main features; (b) detail.

in four surrounding boxes, copy the square (but not its internal structure) at the corners of a rectangle, and connect the rectangle with the corners of the surrounding boxes.

The main features of this construction are shown in Figure 4(a). However we must surround the square with some machinery in order to achieve no obtuse angles. In particular we form a small grid of rectangles and triangles, shown in Figure 4(b). The triangle sides are tangent to a circle centered on the opposite corner of the surrounding box, so the triangles formed by connecting those sides to the opposite corner are nonobtuse. The grid itself can be triangulated by right triangles.

Theorem 3. *For any point set X, there is a triangulation containing the points of X as vertices, with no obtuse angles, and with size $O(|X|)$.* ∎

4. Only acute angles

We have shown that triangulations with aspect ratio at most 2, and maximum angle at most $90°$, are possible. A natural question is how far these bounds can be extended. The two bounds are closely interlinked; a triangle with aspect ratio bounded below 2 has maximum angle bounded below $90°$, and conversely any triangle with maximum angle bounded below $90° - \epsilon$ has bounded aspect ratio. In this section, we first show that the aspect ratio bound can be reduced to $5/3$; triangles with this aspect ratio have maximum angle at most $80°$. Our triangulation also achieves a minimum angle of at least $36°$. Second, we extend our linear-size nonobtuse triangulation algorithm to one that finds a triangulation in which all angles are strictly acute. Of course, some angles may be arbitrarily close to $90°$, for otherwise we would have a bound on the aspect ratio.

Recall that, in our previous constructions, we triangulated unwarped boxes with isosceles right triangles. Clearly, this must be changed to improve the aspect ratio beyond 2. Indeed, the main difficulty is triangulating unwarped boxes; the boxes near input points can be dealt with by grid-shifting and special constructions analogous to those above.

Thus we first consider an unwarped quadtree. In the previous constructions we imposed a *balance* condition on the quadtree, namely, that no unsplit box is orthogonally adjacent to a box with 1/4 its side length. Now we need a somewhat stricter condition: diagonally

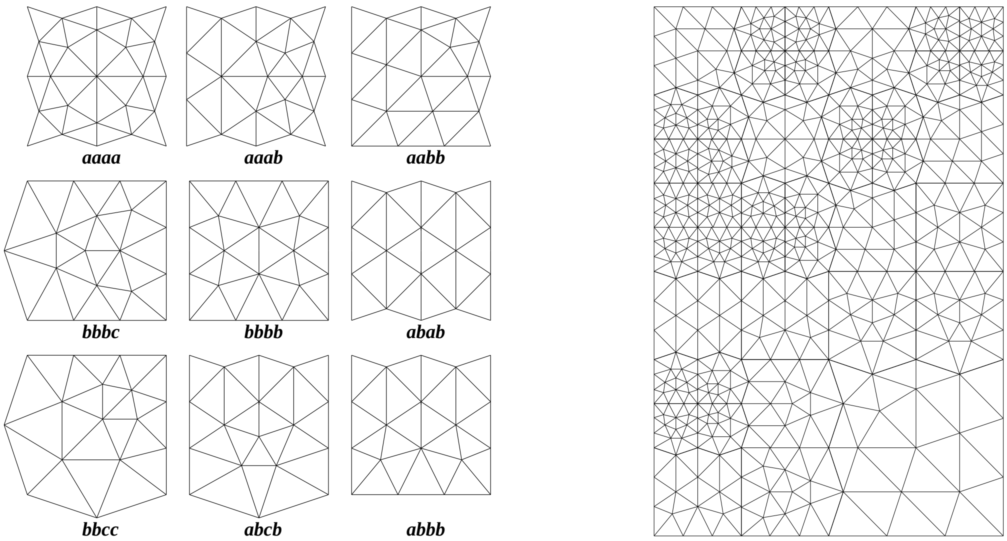

Figure 5. Acute triangulation: (a) tiles for labeled squares; (b) example triangulation.

adjacent boxes must also be within a factor of two in size. This does not affect the number of boxes by more than a constant. We say a quadtree satisfying this condition is *strongly balanced*.

We assign labels from the set $\{a, b, c\}$ to the sides of each unsplit box. If a box is orthogonally adjacent to two smaller boxes, the corresponding side is labeled a. If a box is adjacent to one its own size, the side is labeled b. And if a box is adjacent to a larger box, the side is labeled c. This causes b labels to be matched opposite other b's, and a labels to be matched opposite pairs of c's. Given a labeled box, we describe its labeling by writing the edge labels clockwise starting from the top; for example, *abbb* would be a box with the top side subdivided, and the other sides adjacent to boxes of the same size.

Lemma 4. *All boxes in a strongly balanced quadtree are labeled with a reflection or rotation of one of the following nine label patterns: aaaa, abbb, aabb, abab, aaab, abcb, bbbb, bbcc, and bbbc.*

Proof: Most other possible patterns contain an a label adjacent to a c label, which cannot happen because of the balance condition. The remaining cases have a c opposite another c. This would imply two larger neighbors separated by half their side length, an impossibility in a quadtree. ∎

Now we deform each box (with side length ℓ) as follows. Each side labeled c is split into two equal-length segments, which project out a distance of $\ell/6$ from the square of the box. Each side labeled a is split into four equal-length segments, which project into the square, matching the projections on the corresponding half-size sides labeled c. Finally, each side labeled b is split into three equal-length segments, running in a line along the side of the square.

Lemma 5. *Each deformed box can be triangulated with maximum aspect ratio* $5/3$.

Proof: Tiles for the patterns of Lemma 4 are depicted in Figure 5(a). The aspect ratio bound follows from a tedious calculation on each of the triangles in each of the tiles. ∎

Thus it follows that any balanced quadtree can be triangulated with aspect ratio 5/3. An example of how differently sized copies of the tiles in Figure 5(a) fit together to make a triangulation is shown in Figure 5(b).

Theorem 4. *For any point set X, there is a triangulation containing the points of X as vertices, with no obtuse angle, with aspect ratio at most $5/3$, and with size $O(|\mathcal{QT}(X)|)$.*

Proof: We use a strongly balanced quadtree, and triangulate unwarped boxes using the labels and tiles described above. It remains to show how to warp the quadtree boxes to fit the input points. As before, we use the grid shifting technique. With a large enough construction, we can use this technique to force each input point to be within a square that is as small as we desire relative to its surrounding box, and that lies at any desired location in that box. Thus, all we need is a tile such that, if one of its interior points is moved within a small neighborhood, all aspect ratios remain no larger than 5/3. Equivalently, the interior point must be adjacent to triangles with aspect ratios all strictly less than 5/3. The center point of the *aaaa* tile, as drawn in Figure 5(a), is adjacent to triangles with aspect ratio 8/5, and so satisfies this condition. ∎

It seems likely that a similar algorithm, with a stronger balance condition and a more complicated labeling system, can achieve improved aspect ratio and angle bounds. In particular, it might be possible to construct optimal size triangulations with maximum angle 72°. Further improvements would be more difficult, as they would force all internal vertices to have degree six or more. It is also reasonable to consider improving the minimum angle of our triangulations. The construction used in Theorem 4 gives a minimum angle of 36.87° (a little better than the angle implied by aspect ratio 5/3), and the construction of Theorem 2 can be modified to achieve a minimum angle of $45° - \epsilon$, for any $\epsilon > 0$. But again it seems likely that more complicated constructions can achieve minimum angle 51.4°. Again, any further improvement would be difficult, as such an improvement would eliminate interior vertices with degree seven or more.

Finally, we consider possible improvements to Theorem 3. There is little to do; bounding all angles below $90° - \epsilon$, for constant ϵ, would imply bounded aspect ratio and nonlinear size for some inputs. However our previous construction includes many right triangles; we now modify it so that all triangles are strictly acute.

Theorem 5. *For any point set X, there is a triangulation containing the points of X as vertices, with no obtuse or right angles, and with size $O(|X|)$.*

Proof: Our algorithm is as before, but using the strongly balanced quadtree and the labeled tiles of Figure 5(a). There are two problems to solve. First, we must connect these tiles to the gadget depicted in Figure 4(a). Second, we must cause all triangles in the gadget to be acute, rather than right.

The first problem is solved as follows. The gadget of Figure 4(a) lies in a square with sides subdivided into two equal segments. We have no tiles of this kind; however, we can modify the *bbbb* tile, so that it has such a square in its center. More precisely, let s be a square with sides divided into three equal segments, as in the *bbbb* tile. Let s' be a square

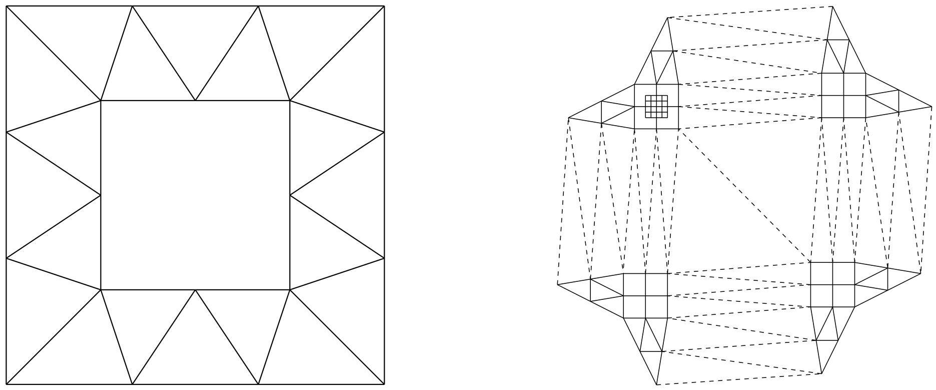

Figure 6. Linear-size acute triangulation: (a) connecting tile; (b) moving corners out.

concentric with s, but with half the side length, and with sides divided into two equal segments. Then the space between s and s' can be triangulated with acute triangles of aspect ratio at most 3/2, as in Figure 6(a).

As before, we can use grid shifting to move the small cluster into any desired position relative to the square in Figure 6(a). We assume that the cluster is triangulated by a quadtree with all outside box sides labeled b. By choosing an appropriately-sized root box for the cluster, we can arrange that there will be exactly 8 box sides along each side of the root box. Then the three copies of the cluster that are symmetrically placed in the gadget of Figure 4(a) can be made from 64 *bbbb* tiles. Recall that each copy of the cluster lies inside a roughly triangular section of "gridwork" abutting a circular arc of segments; in Figure 4(b) the gridwork comprises all the rectangles and triangles that do not run off the edge of the figure. By slightly tilting all the vertical and horizontal line segments in the gridwork towards the closest corner of the large surrounding box (that is, the verticals in Figure 4(b) tilt from northwest to southeast), we can make all the triangles in the gridwork acute. The rectangles become parallelograms that can be triangulated by acute triangles.

Finally, between each copy of the cluster and in the center of the whole construction, there are a number of rectangles. We move two copies of the cluster outwards, so that these rectangles become parallelegrams that can be triangulated by acute triangles. This rearrangement, and the gridwork distortion mentioned above, are depicted in Figure 6(b). As drawn, there are some obtuse angles, because the gridwork distortion is not to scale; if the distortion were made sufficiently small, all angles would be acute. ∎

5. Segments and polygons

In this section, we generalize the point set input first to a set of nonintersecting line segments and then to a polygonal region with polygonal holes. A triangulation $\mathcal{T}$ *respects* the input if each vertex of the input is a vertex of $\mathcal{T}$, and each nondegenerate edge of the input is a union of edges of $\mathcal{T}$. For line segment input, $\mathcal{T}$ is a triangulation of a convex polygon covering the input; for a polygonal region, all triangles of $\mathcal{T}$ must lie within the input region. In each case, we seek a triangulation with bounded aspect ratio that respects the input.

5.1. Nonintersecting segments

Let S be a set of line segments that do not intersect even at endpoints, and let X be the set of endpoints of segments in S. For a point x of a segment, the *nearest foreign neighbor* of x is the closest point of a different segment. A quadtree box b of side ℓ is *crowded* if one of the following holds.

C1. Box b contains a member of X whose nearest neighbor in X is as close as $2\sqrt{2}\,\ell$.

C2. Box b contains a member of X and one of the extended neighbors of b is split.

C3. Box b contains a point x of a segment of S and the nearest foreign neighbor of x is as close as $2\sqrt{2}\,\ell$.

As in the basic algorithm, we can start with any root box that has side length only a constant factor times the diameter of the input, and that places all input segments well away from its boundary. For example, we may start with a root box twice the size and concentric with the minimum bounding square. As in Section 4, we impose stronger balance conditions: no leaf box may be orthogonally adjacent to one more than twice its size, nor may it be both adjacent (diagonally or orthogonally) to one twice its size and orthogonally adjacent to one half its size. Each leaf box containing a point of X must be surrounded by 24 boxes (i.e., two layers) its own size. These conditions simplify the analysis and improve the aspect ratio, while changing the size by only a constant factor.

A *q-vertex* is the point at which a segment of S crosses a quadtree box boundary. An *edge* of the quadtree subdivision is an edge of its graph structure; thus a split side of a leaf box is a path of two edges. A *side* is a maximal segment along the boundary of a polygon. We warp the quadtree to fit S in the following steps.

1. Each point of X chooses its closest quadtree vertex, and we replace each chosen vertex with the (unique) endpoint that chose it. This destroys q-vertices on edges incident to a chosen vertex.

2. Next each remaining q-vertex chooses its closest quadtree vertex that has not yet moved, and we warp chosen vertices to their choosing segments. With one exception, we warp vertically (that is, along a vertical trajectory) to segments with slope in the range $[-1, 1]$ and horizontally to other segments. The exception is that when a corner of an already-warped box is chosen only once, we warp it to its chooser.

3. Now we have two rules involving split sides. As in step 2, vertices move horizontally or vertically depending on the segment's slope.

   (a) If the two endpoints of a split side of a box both warped to a segment s in step 2, then we also warp the midpoint to s if we have not already done so.

   (b) If a split side of a box is crossed by segment s, then both endpoints of the crossed edge must warp to s. We now warp such an endpoint (corner or midpoint) to s if we have not already done so.

Each face in the planar subdivision is then triangulated by first choosing the diagonals that lie along segments of S and then choosing the remaining diagonals that give the best

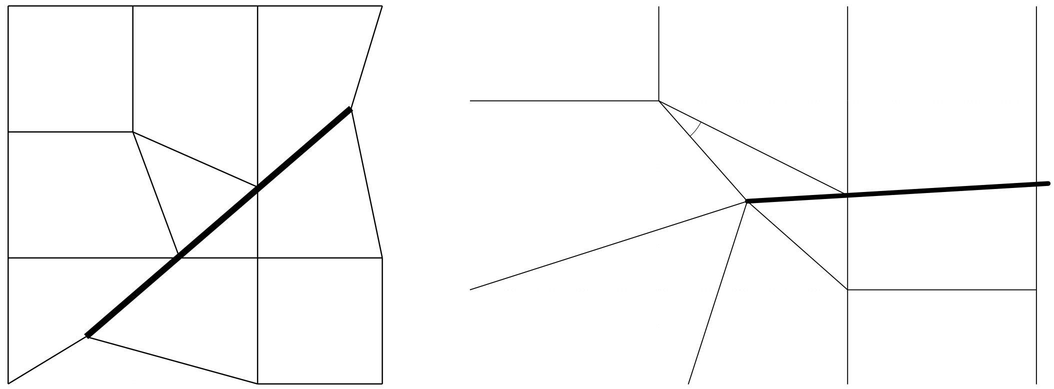

Figure 7. The warped quadtree framework for a segment: (a) typical case; (b) worst-case angle.

aspect ratio. The resulting triangulation is denoted $\mathcal{QT}(S)$. Figure 7(a) shows the warped quadtree for a single segment input. The upper right corner of the lower left box is an example of the exception in step 2.

Lemma 6. *$\mathcal{QT}(S)$ respects S.*

Proof: Each member of X chooses a quadtree vertex to be warped to it, and no quadtree vertex is chosen by two distinct members of X. In the second warping step, each edge of the quadtree that is crossed by a segment s warps so that at least one of its endpoints lies on s. This destroys all q-vertices. The third step does not introduce new q-vertices, so the interior of a warped box contains a point of a segment s only if s crosses the box as a diagonal. ∎

Lemma 7. *For all S, $A(\mathcal{QT}(S)) \leq 5$ and the minimum angle in $\mathcal{QT}(S)$ measures at least* $18.4°$.

Proof: The proof involves a rather tedious case analysis, so we omit some details. Let b be a box of side ℓ in the original unwarped quadtree subdivision. Let b' be the warped counterpart of b. Vertices of b' lie either in their original locations or along a segment $s \in S$.

There are three cases. The first case is: b is surrounded by eight boxes its own size. In this case, at most two vertices of b warp. Now there are two subcases, depending on whether a vertex of b' lies at a member of X or not. If not, then two vertices of b that warp both move in the same direction (i.e., either horizontally or vertically), thus maintaining their original distance from each other. All edges along the boundary of b' have lengths between $\ell/2$ and $3\ell/2$, and it is not hard to confirm that all angles (between adjacent sides of b' or between a side of b' and s) measure at least $\arctan(1/2) > 26.5°$. If there is a member of X, the worst case occurs when the vertex lies at the center of b, as shown in Figure 7(b). The sharpest angle is then $45° - \arctan(1/2) = \arctan(1/3) > 18.4°$, and the aspect ratio of a triangle with angles $\arctan(1/2)$ and $\arctan(1/3)$ is 5.

The second case is: b has no split sides, but is adjacent (orthogonally or diagonally) to a box twice its own size. Edges along the boundary of b' have lengths between $\ell/2$ and 2ℓ, but the ratio of longest to shortest is again no more than 3, since vertices of b all move horizontally or all move vertically. Notice that warping step 3(b) may reduce b' to only three sides. Again all angles measure at least 26.5°.

The last, and most complicated, case is: b has at least one split side. Arbitrarily small angles may arise between sides of the warped box and s when the two endpoints, but not the midpoint, of a split side warp. Warping step 3(a) removes these angles. Warping step 3(b) guarantees that all edges of b' have lengths between $\ell/2$ and $3\ell/2$. Notice that two vertices of b that both warp to s either maintain (at least) their original distance apart or coalesce (reducing to the case of an unsplit side). All angles in b' and between sides of b' and s turn out to be at least 18.4°. The triangulation of b' can be completed with angles no smaller than 18.4°. ∎

Let $\mathcal{CDT}(S)$ denote the *constrained Delaunay triangulation* [9, 18] of S. Each segment of S is an edge of $\mathcal{CDT}(S)$, and another edge e between vertices of X appears in $\mathcal{CDT}(S)$ iff there is an "empty" circumcircle of e. A circumcircle is empty if each vertex of X in its interior is not visible to one of the endpoints of e. $\mathcal{CDT}(S)$ maximizes the minimum angle among all triangulations that respect S and add no new vertices [18].

Lemma 8. *$|\mathcal{QT}(S)|$ is $O(\sum A(a,b,c))$, where the sum is over all triangles abc in $\mathcal{CDT}(S)$.*

Proof: As in the proof of Lemma 2, the size increase due to the balance condition is amortized against crowded boxes. The number of boxes that are crowded due only to C2 is linear in $|X|$. Lemma 2 bounds the number of boxes crowded (by C1) because they contain both endpoints of a single segment.

Condition C3 requires segments to be well separated. As an example, consider two closely spaced parallel segments e and f. The quadtree will split until segment e intersects boxes of side length about one-fourth the distance between e and f. The number of such boxes is bounded by a constant times the aspect ratio of a triangle with base e and apex at one of the vertices of f. One of the two such triangles must be a triangle of $\mathcal{CDT}(S)$.

In general, let b be a box that is crowded because it contains a point of a segment e and some nearby box (up to two away) contains points of another segment f. Consider the four triangles in $\mathcal{CDT}(S)$ that are supported by either e or f, and charge b's split to the one with minimum altitude. Thus each triangle in $\mathcal{CDT}(S)$ is charged only by boxes of side length at least a constant fraction of its altitude. Since b must also be within two boxes of a side of the triangle it charges, each triangle of $\mathcal{CDT}(S)$ is charged by a number of boxes proportional to its aspect ratio. ∎

Theorem 6. *Suppose S is a set of strictly nonintersecting segments and $\mathcal{T}$ is any triangulation respecting S. The quadtree method produces a triangulation $\mathcal{QT}(S)$ respecting S, with aspect ratio at most 5, and with size $O(|\mathcal{T}|A(\mathcal{T}))$.*

Proof: We generalize the line-segment problem somewhat to allow an input that includes subdivided line segments, that is, segments with vertices in the middle. A triangulation must include these vertices as well. The quadtree algorithm remains unchanged, though now X must be interpreted as all vertices, and "segment" means an entire straight chain. Lemma 8 follows exactly as above.

Triangulation $\mathcal{T}$ subdivides the segments of S in some way. Let S' be the segments of S, subdivided according to $\mathcal{T}$, along with the other vertices of $\mathcal{T}$ included as zero-length segments. As in the case of point set input, the quadtree algorithm (with a suitable root box) is monotonic; that is, if $S \subseteq S'$, then $|\mathcal{QT}(S)| \leq |\mathcal{QT}(S')|$. Now the bound on the size of $\mathcal{QT}(S)$ follows from Lemma 8 and $A(\mathcal{CDT}(S')) \leq 2A(\mathcal{T})$. ∎

Corollary 3. *For $\alpha \geq 5$, let $OPT_\alpha(S)$ be the minimum size of a triangulation respecting S with aspect ratio at most α. Then there is a constant c_α such that for all S, $|\mathcal{QT}(S)| \leq c_\alpha \cdot OPT_\alpha(S)$.* ∎

5.2. Polygonal regions

Now we generalize the input to a closed polygonal region P with polygonal, possibly degenerate, holes. We initially assume that no angle of ∂P facing the interior of P is acute. Later we relax this restriction to an arbitrary, fixed lower bound.

Let x be a point of ∂P. Point y of ∂P is *foreign* to x if y is on another connected component of ∂P, or if every walk from x to y along ∂P includes at least two vertices. The endpoints of the walk count; thus all vertices of ∂P are foreign to each other. The *nearest foreign neighbor* of x is the closest point of ∂P foreign to x, where distance is now *geodesic distance*. (The geodesic distance between two points in P is the length of a shortest path between them that lies entirely in P.) A quadtree box b of side ℓ is *crowded* if one of the following holds, where X now denotes vertices of ∂P.

C1. Box b contains a point x of ∂P and the nearest foreign neighbor of x is as close as $2\sqrt{2}\,\ell$.

C2. Box b contains a point of X and one of the extended neighbors of b is split.

As in previous sections we recursively split crowded boxes and propagate these splits. We impose the same balance and 24-neighbor conditions as for segments. Further assume that no member of X lies exactly in the center of a box. Again a q-vertex is an intersection of an edge of ∂P and a box boundary. In degenerate cases, a single point may be the site of two different q-vertices, for example, in the case of a line-segment hole.

We now describe how to warp the quadtree subdivision to fit P. There is an added complication in this warping procedure: a single quadtree box b may contain vertices in more than one connected component of $P \cap b$. Roughly speaking, we warp b separately for each connected component.

For a quadtree vertex y let B_y denote the union of the three or four boxes whose boundaries contain y. In the warping steps below, distance is Euclidean distance in the plane, not geodesic distance.

1. Each member of X and each q-vertex chooses its closest quadtree vertex. We "split" a vertex y that is chosen by at least one member of X into at most four copies. We warp a copy of y to each member of X that chooses it, and to the closest q-vertex choosing y in a connected component of $P \cap B_y$ that contains no member of X. (If y is chosen only by q-vertices, then we do not move it yet.)

2. Next each remaining q-vertex chooses its closest quadtree vertex that has not yet moved. We warp a copy of a chosen vertex y to each edge of ∂P containing a q-vertex that chose y. (We show below that there will be at most one such edge for

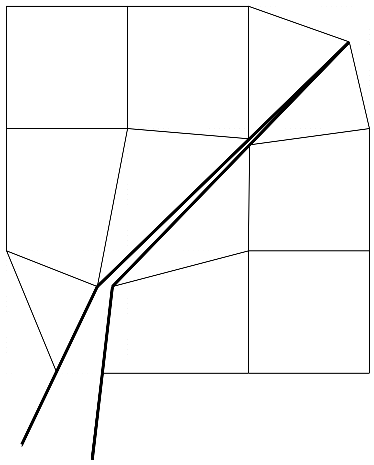

Figure 8. Warping to a polygon.

each connected component of $P \cap B'_y$, where B'_y is the current warped version of B_y.) Vertices move horizontally or vertically exactly as in the case of segments.

3. We again have the two rules involving split sides.

   (a) If the two endpoints of a split side of a box both moved to an edge s of ∂P in step 2, then we must also warp the midpoint of that side to s.

   (b) If a split side of a box is crossed by an edge s of ∂P, then we must warp both endpoints of the crossed edge to s.

For example, step 1 moves a copy of the upper right corner of the lower left box to each of two vertices of ∂P in Figure 8. Step 2 splits the lower left corner of the upper right box and warps these copies to two q-vertices. Edges crossing ∂P are removed, but edges that do not cross ∂P remain.

After the warping steps, we remove vertices and quadtree edges contained in the complement of P. Finally, we triangulate faces of the warped quadtree subdivsion by choosing all diagonals lying along ∂P and the remaining diagonals that give the best aspect ratio. The resulting triangulation is denoted $\mathcal{QT}(P)$.

Lemma 9. *$\mathcal{QT}(P)$ respects P.*

Proof: After step 1 above, each vertex of ∂P coincides with a quadtree box corner. Suppose that in step 2 a quadtree vertex y is chosen by two distinct q-vertices p and q. We assert that either p and q are in separate connected components of $P \cap B'_y$ or p and q lie on the same edge of ∂P. Assume the contrary. Then p and q must lie on adjacent edges of ∂P, or else they would be foreign to each other and the quadtree would have refined further. So let v be the vertex of ∂P between p and q. Our assumptions imply that v is nearer to y than to any other quadtree vertex. Hence y should have warped to v in step 1, destroying p and q, a contradiction. It is now straightforward to confirm that after step 2, all q-vertices have disappeared. ∎

Lemma 10. *$A(\mathcal{QT}(P)) \le 5$, and the minimum angle in $\mathcal{QT}(P)$ measures at least $18.4°$.*

Proof: Let f' be a face in warped box b' after all warping steps have taken place, but before any diagonals have been chosen. The vertices of f' are all warped copies of the vertices of some box b. Face f' is bounded by at most two edges lying along ∂P. If f' is bounded by fewer than two such edges, then face f' could have arisen in the case of line segment input, so the aspect ratio is bounded by Lemma 7.

If f' is bounded by two edges s_1 and s_2 lying along ∂P, then vertex v—the meeting point of s_1 and s_2—lies on the boundary of f', and b was a box surrounded by eight boxes its own size. First assume that v lies inside b, and without loss of generality, in the upper-left quarter of b. Then we distinguish a number of cases, depending upon which corners of b are closest to the q-vertices at which s_1 and s_2 cross the boundary of b. The assumption that the angle between s_1 and s_2 is at least $90°$ makes this case analysis quite easy. Second, assume that v lies outside b, and a corner of b warped out to it. Again the bounds follow by a straightforward case analysis. ∎

Theorem 7. *Suppose P is a polygonal region (with holes) in which no interior angle is acute, and $\mathcal{T}$ is any triangulation respecting P. The quadtree method produces a triangulation $\mathcal{QT}(P)$ respecting P, with $A(\mathcal{QT}(P)) \le 5$, and with size $O(|\mathcal{T}|A(\mathcal{T}))$.*

Proof: We define $\mathcal{CDT}(P)$ to be the portion of the constrained Delaunay triangulation of ∂P that lies within P. The size bound then follows analogously to the previous arguments. ∎

Finally we consider the general case of polygonal regions with all interior angles greater than β. Our strategy is to reduce the general case to the case just considered by cutting off isosceles triangles containing the acute interior angles; this idea also appears in [2].

Suppose there is an acute interior angle of P with vertex v. We grow the quadtree just as if P had no acute interior angles. Vertex v ends up in a leaf box b surrounded by eight neighbors its own size. We cut off the largest isosceles triangle with legs along ∂P and apex v that fits inside the union of these nine boxes. This introduces a new side, called a *cut side*, to P. We do the same for each acute interior angle of P. This leaves a polygon P' with no acute interior angles that can be triangulated by the method above. Where they overlap, the quadtree subdivision for P' is a refinement of the one for P, and we have added only $O(1)$ boxes per cut side, with the exact constant depending upon β. For simplicity we further subdivide until all boxes intersecting any one cut side are the same size. Then in the warping step, the quadtree vertices that warp to a cut side subdivide the cut side into equal-length edges, except for the first two and last two edges. The size $|\mathcal{QT}(P')|$ is $O(|\mathcal{T}|A(\mathcal{T}))$ for any triangulation $\mathcal{T}$ of P.

It remains to triangulate the isosceles triangles in a way that is compatible with $\mathcal{QT}(P')$. Assume we are given an isosceles triangle I with an acute angle at its apex and a base that is subdivided into some number of edges with endpoints $v_1, v_2, \ldots, v_m$. Further assume that all base edges $v_i v_{i+1}$, except $v_1 v_2$, $v_2 v_3$, $v_{m-2} v_{m-1}$, and $v_{m-1} v_m$, have the same length ℓ and that the lengths of the exceptional edges are in the range $[\ell/2, 3\ell/2]$. We now show how to compute a linear-size, bounded-aspect-ratio triangulation of I.

If the base s of I consists of a single edge then we are done. Otherwise we gather the vertices along s into overlapping groups of three: $G_1 = \{v_1, v_2, v_3\}$, $G_2 = \{v_3, v_4, v_5\}$, and

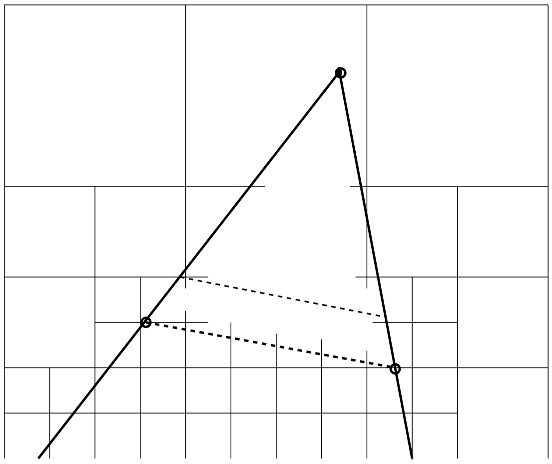

Figure 9. An acute interior angle.

so forth. There may be one group of only two at the end. We choose the line segment e, parallel to s and distance ℓ closer to the apex of I, as shown in Figure 9. For each group G_i, except the first and last, we place a vertex u_i along e perpendicularly across from the middle member of G_i. These vertices are distance 2ℓ apart. The first and last u_i vertices are the endpoints of e. If the first segment u_1u_2 has length greater than 3ℓ, we add a vertex at distance 2ℓ from u_2; we treat the last segment similarly.

We triangulate the trapezoid between s and e by adding edges between members of G_i and u_i, except for the first and last groups. We complete the triangulation at the beginning and end of e with diagonals giving best aspect ratio. We then recursively triangulate the triangle with base e.

Theorem 8. *Suppose P is a polygonal region (with holes) in which each interior angle measures at least β. Let $\mathcal{T}$ be any triangulation respecting P. The quadtree method produces a triangulation $\mathcal{QT}(P)$ respecting P, with aspect ratio at most* $\max\{5, 1/\sin\beta\}$ *and minimum angle at least* $\min\{18.4°, \beta\}$*, and with size $O(|\mathcal{T}|A(\mathcal{T}))$.* ∎

Corollary 4. *Let $OPT_\alpha(P)$ be the minimum size of a triangulation respecting P with aspect ratio no more than $\alpha \geq \max\{5, 1/\sin\beta\}$. There is a constant c_α such that for all P, $|\mathcal{QT}(P)| \leq c_\alpha \cdot OPT_\alpha(P)$.* ∎

6. Dimensions 3 and above

A triangulation in $d \geq 3$ dimensions is a partition into d-simplices. The quadtree algorithm of Section 2 extends immediately to a 2^d-tree algorithm for general dimension.

Theorem 9. *Suppose X is a point set in $\mathcal{R}^d$ and $\mathcal{T}$ is a triangulation that respects X. Then there is a triangulation $\mathcal{QT}(X)$ that respects X and has bounded aspect ratio and size $O(|\mathcal{T}|\log A(\mathcal{T}))$.*

Proof: The construction follows that of Theorem 1, refining a balanced 2^d-tree until each box with a point is surrounded by $3^d - 1$ empty boxes the same size, moving the nearest box corner to the point, and finally dividing each box into simplices.

The analysis differs from the planar case because the Delaunay triangulation may not be within a constant of minimum aspect ratio. Instead, we bound $|\mathcal{QT}(X)|$ by $O(\sum \log A(t))$, where t ranges over the simplices of an arbitrary triangulation $\mathcal{T}$. As in Lemma 2, the only nonlinear behavior occurs when a crowded box is split repeatedly without separating a cluster.

If any two points of the cluster are adjacent in $\mathcal{T}$, then some two cluster points and one non-cluster point are the vertices of a two-dimensional triangle in $\mathcal{T}$. That triangle has large aspect ratio, and therefore so do the simplices of $\mathcal{T}$ that it bounds. We charge the size increase for the cluster to one of those simplices.

If on the other hand no cluster points are adjacent, let y_1 and y_2 be any two cluster points. Since line segment $y_1 y_2$ does not lie within a single simplex, there is a simplex $t \in \mathcal{T}$ with $d-1$ facets meeting in a vertex at y_1 for which $y_1 y_2$ intersects the d-th facet. Some altitude of t is at most the diameter of the cluster, and the other vertices of t are all outside the cluster, so t has aspect ratio at least the cluster's distance from its nearest neighbor divided by its diameter. We charge the size increase for the cluster to t. ∎

Corollary 5. *Let $OPT_\alpha(X)$ be the minimum size of a triangulation of the point set X achieving aspect ratio α. For each sufficiently large α, there is a constant c_α such that $|\mathcal{QT}(X)| \leq c_\alpha \cdot OPT_\alpha(X)$.* ∎

Theorem 10. *Fix a dimension d, and let X be a point set in $\mathcal{R}^d$. Then there is a set $Y \supseteq X$ of $O(|X|)$ points for which the d-dimensional Delaunay triangulation contains only $O(|X|)$ simplices.*

Proof: The proof is similar to that of Theorem 3. We refine a balanced tree until each box with a point is surrounded by empty boxes the same size, move the nearest box corner to each point, and take Y to be the set of box corners. As in Theorem 3, when a constant number of splits fail to separate a cluster, we triangulate the cluster recursively. By using a somewhat larger root box for the cluster, we guarantee that every d-sphere that contains both cluster points and non-cluster points has one of the new points in its interior. Then every point is incident on a bounded number of maximal empty spheres, so the Delaunay triangulation has bounded degree. ∎

7. Conclusions

We have shown how to generate triangular meshes of guaranteed quality and size for several classes of input and two measures of quality. We have also shown that a planar point set admits a linear-size acute triangulation, and a d-dimensional point set admits a linear-size "Steiner Delaunay" triangulation. The key points of the quadtree refinement algorithms are keeping the tree locally balanced (at a constant factor in amortized cost), and either charging the cost of narrow parts of the tree to expensive features of the input, or skipping over them altogether with constructions of constant cost.

Subsequent to the work reported here, Eppstein [14] showed how to use essentially the same algorithm as that of Theorem 1 to approximate the minimum-total-length Steiner triangulation of a point set. Bern and Eppstein [5] devised an algorithm for triangulating an n-sided polygonal domain with $O(n^2)$ nonobtuse triangles; Melissaratos and Souvaine [21] merged our methods with those of Baker et al. to produce triangulations with no small and no obtuse angles; and Mitchell and Vavasis [23] showed how to use octrees to triangulate polyhedra with bounded-aspect-ratio tetrahedra. Bern and Eppstein [6] survey recent work in computational geometry motivated by mesh generation.

There remain several avenues for further research. The first is to extend our quadtree methods to more complicated inputs: triangulating a planar straight line graph (in which vertices may have degree greater than two), and tetrahedralizing polyhedral cell complexes. These problems have application to domains composed of more than one material.

Second, it would be nice to reduce the constant factors in our algorithms, both in the aspect ratio and—especially—in the mesh size. As Figure 2(b) shows, simple heuristics can be effective in removing unnecessary points from the triangulations our algorithm produces. Using binary trees of rectangles with aspect ratio $\sqrt{2}$ in place of quadtrees might also improve the size.

Finally, there seem to be some "threshold phenomena" worth investigating. As we have shown, triangulations with angles at most 90° require only linear size. Angles bounded below 90° imply bounded aspect ratio, and hence nonlinear worst case size. We believe a further barrier in triangulation difficulty occurs at maximum angle 72°, or minimum angle 51.4°, beyond which all vertices except those near the boundary must have exactly six neighbors. It seems that triangulations with angles arbitrarily close to 60° can always be found, but that they may require many more triangles than we used in our constructions. However we have not proved any upper or lower bounds for this case.

Acknowledgements

We thank Randy Bank, Paul Chew, Eric Grosse, Rob Schreiber, and Warren Smith for discussions of mesh generation in theory and practice; Dennis Jespersen and Steve Hammond for Figure 1; and David Dobkin for bringing the no-obtuse-angles problem to our attention.

References

[1] I. Babuška and A. K. Aziz. On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis*, 13:214–226, 1976.

[2] Brenda S. Baker, Eric Grosse, and Conor S. Rafferty. Nonobtuse triangulation of polygons. *Discrete and Computational Geometry*, 3:147–168, 1988.

[3] R. E. Bank. *PLTMG User's Guide*. SIAM, 1990.

[4] Timothy J. Barth and Dennis C. Jespersen. The design and application of upwind schemes on unstructured meshes. In *27th Aerospace Sciences Meeting*. AIAA, 1989.

[5] Marshall Bern and David Eppstein. Polynomial-size nonobtuse triangulation of polygons. In *Proceedings of the 7th Symposium on Computational Geometry*, pages 342–350. ACM, 1991.

[6] Marshall Bern and David Eppstein. Mesh Generation and Optimal Triangulation. In *Computing in Euclidean Geometry*, World Scientific, to appear in 1992.

[7] B. Chazelle, H. Edelsbrunner, L. J. Guibas, J. E. Hershberger, R. Seidel, and M. Sharir. Selecting multiply covered points and reducing the size of Delaunay triangulations. In *Proceedings of the 6th Symposium on Computational Geometry*, pages 116–127. ACM, 1990.

[8] Bernard Chazelle and Leonidas Palios. Triangulating a nonconvex polytope. *Discrete and Computational Geometry*, 5:505–526, 1990.

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

[10] L. Paul Chew. Guaranteed-quality triangular meshes. Technical Report TR–89–983, Cornell University, 1989.

[11] E. F. D'Azevedo and R. B. Simpson. On optimal interpolation triangle incidences. *SIAM Journal on Scientific and Statistical Computing* 10: 1063–1075.

[12] Herbert Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer-Verlag, 1987.

[13] H. Edelsbrunner, T. S. Tan, and R. Waupotitsch. A polynomial time algorithm for the minmax angle triangulation. In *Proceedings of the 6th Symposium on Computational Geometry*, pages 44–52. ACM, 1990.

[14] David Eppstein. Approximating the minimum weight triangulation. In *Proceedings of the 3rd ACM-SIAM Symposium on Discrete Algorithms*, 1992.

[15] I. Fried. Condition of finite element matrices generated from nonuniform meshes. *AIAA Journal*, 10:219–221, 1972.

[16] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[17] Barry Joe. *Finite element triangulation of complex regions using computational geometry.* PhD thesis, University of Waterloo, 1984.

[18] D. T. Lee. Generalized Delaunay triangulation for planar graphs. *Discrete and Computational Geometry*, 1:201–217, 1986.

[19] Richard J. Lipton, Donald J. Rose, and Robert Endre Tarjan. Generalized nested dissection. *SIAM Journal on Numerical Analysis*, 16:346–358, 1979.

[20] E. L. Lloyd. On triangulations of a set of points in the plane. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, pages 228–240. IEEE, 1977.

[21] E. Melissaratos and D. Souvaine. Coping with inconsistencies: a new approach to produce quality triangulations of polygonal domains with holes for the finite element method. Technical Report, Department of Computer Science, Rutgers University, 1991.

[22] Gary L. Miller and William Thurston. Separators in two and three dimensions. In *Proceedings of the 22nd Annual Symposium on Theory of Computing.* ACM, 1990.

[23] Scott Mitchell and Stephen A. Vavasis. Quality mesh generation in three dimensions. Manuscript, Center for Applied Mathematics, Cornell University, 1991.

[24] David M. Mount and Alan Saalfeld. Globally-equiangular triangulations of co-circular points in $O(n \log n)$ time. In *Proceedings of the 4th Symposium on Computational Geometry.* ACM, 1988.

[25] Hanan Samet. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley, 1990.

[26] M. S. Shephard. Approaches to the automatic generation and control of finite element meshes. *Applied Mechanics Reviews*, 41:169–185, 1988.

[27] R. Sibson. Locally equiangular triangulations. *Computer Journal*, 21:243–245, 1978.

[28] Warren D. Smith. *Studies in Discrete and Computational Geometry.* PhD thesis, Princeton University, 1988.

[29] Gilbert Strang and George J. Fix. *An Analysis of the Finite Element Method.* Prentice-Hall, 1973.

[30] W. C. Thacker. A brief review of techniques for generating irregular computational grids. *International Journal on Numerical Methods in Engineering*, 15:1335–1341, 1980.

[31] Mark A. Yerry and Mark S. Shephard. A modified quadtree approach to finite element mesh generation. *IEEE Computer Graphics and Applications*, 3:39–46, 1983.