Tight Degree Bounds for Pseudo-triangulations of Points

Lutz Kettner* David Kirkpatrick† Andrea Mantler* Jack Snoeyink*
Bettina Speckmann† Fumihiko Takeuchi‡

Abstract

We show that every set of n points in general position has a minimum pseudo-triangulation whose maximum vertex degree is five. In addition, we demonstrate that every point set in general position has a minimum pseudo-triangulation whose maximum face degree is four (i.e. each interior face of this pseudo-triangulation has at most four vertices). Both degree bounds are tight. Minimum pseudo-triangulations realizing these bounds (individually but not jointly) can be constructed in $O(n \log n)$ time.

1 Introduction

A *pseudo-triangle* is a planar polygon that has exactly three convex vertices, called *corners,* with internal angles less than π. As illustrated in Figure 1, three concave chains, called *sides,* join the three corners. A *pseudo-triangulation* for a set S of n points in the plane is a partition of the convex hull of S into pseudo-triangles whose vertex set is exactly S.

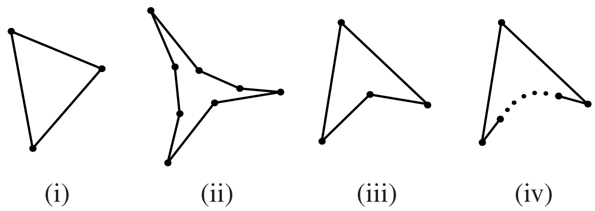


Figure 1: Pseudo-triangles.

Pseudo-triangulations, also called geodesic triangulations, have been studied for convex sets and for simple polygons in the plane because of their applications to visibility [11, 13] and ray shooting [4, 6]. Recently, they have been used in a number of kinetic data structures (KDSs) for collision detection among moving polygons in the plane [1, 8, 9]; more specifically they provide a sparse tessellation of the free space between moving polygons that is easily maintained by edge swaps as polygons move. Furthermore, one-degree-of-freedom mechanisms induced by minimum pseudo-triangulations have been shown to provide efficient primitives for robot arm motion planning problems [15].

*Dept. Computer Science, UNC Chapel Hill, {kettner,mantler,snoeyink}@cs.unc.edu
Research partially supported by NSF grants 9988742 and 0076984.
†Dept. Computer Science, Univ. of British Columbia, {kirk,speckman}@cs.ubc.ca
‡Dept. Information Science, Univ. of Tokyo, fumi@is.s.u-tokyo.ac.jp

We identify the class of *minimum pseudo-triangulations* of a given point set as those that have the minimum number of edges or, equivalently, the minimum number of pseudo-triangle faces (Euler's relation establishes the equivalence). The minimum pseudo-triangulations are of particular interest in the applications above, and also exhibit several geometric and combinatorial properties that make them interesting objects of study in their own right [15, 14].

Since a pseudo-triangulation is a planar graph, we can borrow graph terminology: The *degree of a vertex* is the number of edges incident on it. The *degree of a face* is the number of edges that bound it. Any non-hull edge is adjacent to two *neighboring pseudo-triangles.*

Even though a standard triangulation has average vertex degree $O(1)$, there are sets of n points in the plane for which each possible triangulation contains a vertex of degree $n-1$ (see Figure 1 (iv)). In contrast, we show in Section 3 that every point set in the plane has a minimum pseudo-triangulation of maximum vertex degree 5 (and that such a pseudo triangulation can be constructed for a set of n points in $O(n \log n)$ time). In Section 4 we show that this result is the best possible by demonstrating a set of points for which any pseudo-triangulation has a vertex of degree at least five.

Since minimum pseudo-triangulations have lower average edge degree than general pseudo-triangulations (in particular, triangulations), their average face degree is correspondingly higher. Thus, it is natural to ask whether for every set of points there exists a minimum pseudo-triangulation with a bounded face degree. In Section 5, we describe a construction that yields a minimum pseudo-triangulation in which each face contains either three or four vertices (see Figure 1(i) and 1(iii)). This construction can also be executed for a set of n points in $O(n \log n)$ time.

In the next section we set out some basic properties of minimum pseudo-triangulations. In Section 6 we conclude with some remarks and open problems.

2 Minimum pseudo-triangulations

Let us assume that S is a set of n points in general position. In particular, no three points in S are collinear (degenerate point sets can be handled by applying a symbolic perturbation).

Minimum pseudo-triangulations have several equivalent definitions or characterizations (cf. [15, 12]). In particular, minimum pseudo-triangulations have exactly $n-2$ pseudo-triangles and $2n-3$ edges. Another characterization that relates to the *acyclicity* property described by Streinu [15] is captured in the following:

Lemma 1 *A pseudo-triangulation of S is minimum if and only if every point $p \in S$ has one incident region (either a pseudo-triangle or the exterior face) whose angle at p is greater than π.*

Proof: For a pseudo-triangulation with n vertices and p pseudo-triangles, it follows from Euler's relation that the number of edges $e = (p+1)+n-2$, where unity is added to count the infinite face. The total vertex degree ($2e$) can be bounded from above by considering faces: there are $3p$ corners, and at most n non-corners, since non-corners have angles greater than π. Thus $2e = 2p + 2n - 2 \leq 3p + n$, or $n - 2 \leq p$. Observe that for equality to hold, every vertex must have exactly one angle greater than π. ∎

Every set of n distinct points in the plane has a canonical minimum pseudo-triangulation (referred to as *incremental pseudo-triangulation* in [1]; see Figure 2 (left)). To construct this canonical minimum pseudo-triangulation, first sort the points by x-coordinate, breaking ties by y-coordinate. Then form a triangle with the first three points, and for each subsequent point, add one pseudo-triangle by

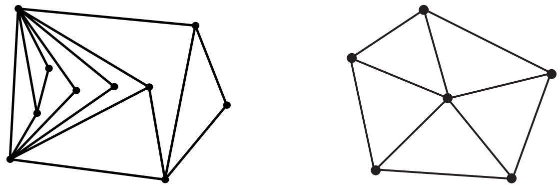


Figure 2: Canonical minimum pseudo-triangulation (left),
minimal but not minimum pseudo-triangulation (right).

creating two tangents to the convex hull. The minimality of the resulting pseudo-triangulation is immediate from Lemma 1.

One cannot always obtain a minimum pseudo-triangulation by simply removing edges. A simple wheel graph, such as a pentagon with one additional vertex in the middle connected to all others (see Figure 2 (right)), is an example for which removing any non-hull edge leaves a convex quadrilateral.

However, for any point set we can always find a minimum pseudo-triangulation that realizes the minimum maximum vertex degree. If the degree is five our construction gives a minimum pseudo-triangulation. If the degree is four a minimal pseudo-triangulation is also minimum as the following lemma shows. In subsequent sections (as the reader will no doubt be delighted to learn) we use the term pseudo-triangulation to refer to a minimum pseudo-triangulation.

Lemma 2 *Any minimal pseudo-triangulation with maximum vertex degree at most* 4 *is a minimum pseudo-triangulation.*

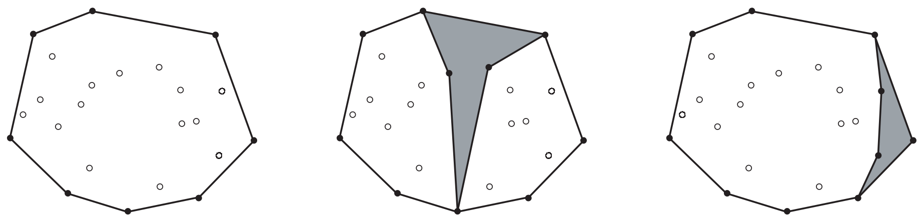
Proof: Assume we are given a pseudo-triangulation of maximum vertex degree at most 4. If the given pseudo-triangulation is not minimum then, by Lemma 1, it must have a point p that does not have an incident region whose angle at p is greater than π. Since p has degree at most 4, we can find a line through p that separates exactly one of its incoming edges from the others. We will now argue that removing this edge e from the pseudo-triangulation will merge its two adjacent pseudo-triangles PT_{left} and PT_{right} into one pseudo-triangle PT. Thus, the given pseudo-triangulation was not minimal.

First of all, the point p is a corner in both PT_{left} and PT_{right} by construction and the removal of e will create an angle greater than π at p in PT. Second, the other endpoint p' of e is a corner in at least one of PT_{left} and PT_{right} and the removal of e will only then create a corner at p' in PT if p' was a corner in both. This implies that the total number of corners in PT is exactly three and hence PT is a pseudo-triangle. ∎

3 Pseudo-triangulations of bounded vertex degree

We describe a recursive construction of a pseudo-triangulation of a set S of n points. We can begin with polygon P as the convex hull of S, since every pseudo-triangulation must use the edges of P. We define the *size* $s(P)$ of a convex polygon P as the number of points of S inside or on the boundary of P. We let $B(P) \subset S$ denote the points on the boundary.

At each step of our recursive construction we apply one of two operations to P to obtain polygons of smaller size. The first operation uses a pseudo-triangle to *partition* P into two convex polygons P_1 and P_2 that share exactly one point p on $B(P)$ and can be separated by a vertical line through p (see Figure 3(i)). This implies that $\max\{s(P_1), s(P_2)\} \leq s(P) - 1$. The second operation *prunes* a point from $B(P)$ and forms a convex polygon P' with $s(P') = s(P) - 1$ (see Figure 3(ii)).



(i) (ii)

Figure 3: The operations: (i) partition and (ii) prune. Note that the shaded pseudo-triangles do not contain any points in their interior.

Now consider a polygon P that describes a subproblem. Since no vertex in the final pseudo-triangulation can have degree greater than five, the incoming edges incident with points on $B(P)$ constrain the choice of pseudo-triangulation edges for the interior of P. We define the *load* of a point p, denoted by $l(p)$, to be the degree of p minus two (for example a point of degree four has a load of two and a point of degree two has no load at all). The load of a polygon P, denoted by $l(P)$, is accordingly defined as the sum of the loads of the vertices on $B(P)$.

We show how to implement the reduction step so as to maintain the following invariants for all polygons P that arise as recursive subproblems:

Invariant 3 *For all $p \in B(P)$, $l(p) \leq 3$. Furthermore, at most one point $p \in B(P)$ has $l(p) = 3$.*

Invariant 4 *If $l(p) \leq 2$ for all $p \in B(P)$, then $l(P) \leq 5$. Otherwise $l(P) \leq 6$.*

Both invariants are trivially true for the convex hull of the set of points.

Let us now assume that we are given a convex polygon P that satisfies both invariants. The appropriate operation to choose depends on the distribution of the load. There are two cases:

1. All points on $B(P)$ have load at most one $\Rightarrow$ **partition**.
2. $B(P)$ has points of load two or three $\Rightarrow$ **prune**.

The following sections explain the two cases and their associated operations in detail and illustrate how the invariants are maintained.

3.1 Partition

Assume we are given a convex polygon P that satisfies both invariants and has boundary points of load at most one. This implies that P contains at most five points of load one on its boundary.

If P contains vertices of load one on its boundary, then we choose a point $p \in B(P)$ with $l(p) = 1$ that has the median x-coordinate among all such points. If P only contains vertices without load on its boundary, then we can choose any point p that is not x-extreme on $B(P)$.

We then split P by a vertical line l_p through p and form two convex polygons P_1 and P_2 that consist of the points of P that lie on the left and on the right of l_p respectively (see Figure 4). Both polygons contain p, which we now regard as a point of load 3.

The polygon P_1 contains exactly one point of load 3, namely p. This satisfies invariant 3 and implies that to satisfy invariant 4, $l(P_1)$ has to be less or equal to 6. This is true, since in addition to p, P_1 contains at most two more points of load 1 on its boundary and the convex hull only contributes an additional load of 1. The same holds with P_1 and P_2 interchanged.

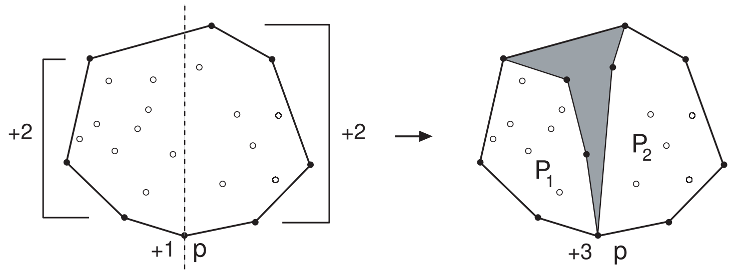


Figure 4: Partitioning the polygon P (each number indicates maximum total load at a point or a chain).

3.2 Prune

Assume we are given a convex polygon P that satisfies both invariants and has at least one point of load two or three.

Let p_{max} be the point on $B(P)$ of the highest load. Let P' denote the polygon formed by pruning p_{max} (see Figure 5). Note that the pseudo-triangle formed by p_{max}, its two neighbours, and part of the convex hull of P' has no points in its interior.

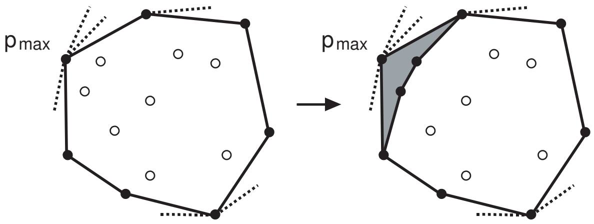


Figure 5: Pruning a high degree vertex.

The load of P' is $l(P') = l(P) - l(p_{max}) + 2 \leq l(P)$, since the new convex hull edges add 2 to the load of P', but p_{max} is a point of load at least 2. Specifically, $l(P') \leq 5$ always holds which implies that invariant 4 is satisfied.

If $l(p_{max}) = 3$ then there was at most one point of load 2 on $B(P)$, all other points must have been of load 1 or less. This implies that at most one of the boundary neighbors of p_{max} had load 2 and therefore at most one point of load 3 is created by the new convex hull edges.

If $l(p_{max}) = 2$ then p_{max} is not necessarily uniquely defined, but invariant 4 implies that in this case there are at most two points of load 2. Since each of them has at least one neighbour of load at most 1, at most one point of load 3 is created by the new convex hull edges.

3.3 Analysis

At this point we have assembled all the necessary facts to prove the following theorem:

Theorem 5 *This recursive construction results in a minimum pseudo-triangulation of vertex degree at most 5.*

Proof: As we have shown in section 3.1 and 3.2, every step of the recursion maintains both invariant 3 and 4. Furthermore the base case of the recursion – a polygon with two points – is trivially pseudo-triangulated. The theorem then follows from invariant 3 and lemma 1. ∎

3.4 Implementation and runtime analysis

To implement our procedure, we would like to maintain convex hulls under the operations of deletion of a point and partitioning by a vertical line. The hull tree of Hershberger and Suri [7] is a variant of the divide-and-conquer hull structure of Overmars and van Leeuwen [10] that supports both operations in amortized $O(\log n)$ time apiece. Recent advances in algorithms for dynamic hull maintenance by Chan [3] and by Brodal and Jacob [2] also support insertion operations, but take slightly more time.

We briefly sketch the idea of the hull tree construction, use, and analysis [7]. Sort the points S by x coordinate, breaking ties by y coordinate. We will store the upper and lower hulls of S in separate hull trees. Construct a binary tree representing the upper hull by a balanced, bottom-up merge: each node stores the common tangent of the upper hulls of its two children. As an accounting device, each point is assigned a number of credits equal to the number of tangents directly above it. Since the tree is balanced, this is at most $\log n$ credits per point.

When a point is deleted, then any tangents that used that point must be recomputed. These tangents can be identified in $O(\log n)$ time by walking down the tree, and can be recomputed in this time plus time proportional to the number of new points that are exposed on the hull at that subtree—this cost can be paid by credits that were assigned when the tree was constructed. When the tree is partitioned by any vertical line, the operation is similar: $O(\log n)$ is spent on the root-to-leaf path to the partitioning line, then tangents are recomputed at a cost that is charged to points exposed to the hull of a subtree.

Theorem 6 *For any set S of n points in the plane, a minimum pseudo-triangulation of S with maximum vertex degree 5 can be computed in $O(n \log n)$ time.*

4 A point set that requires vertex degree 5

This section presents an example that demonstrates that the upper bound presented in the preceding section is the best possible. Specifically,

Lemma 7 *Any pseudo-triangulation of the 12 points of Figure 6 has a vertex of degree at least 5.*

Proof: Suppose, for the sake of deriving a contradiction, that we have a pseudo-triangulation of degree 4. One pseudo-triangle, shown shaded in Figure 6, will use the center point as a non-corner. All other pseudo-triangles will be triangles, since no other internal angles are greater than π. The bound on vertex degree implies that at most three (pseudo-)triangles can use the center as a corner, and that at most three vertices can be between the two corners of the shaded pseudo-triangle that are adjacent to the center point, as shown in the figure.

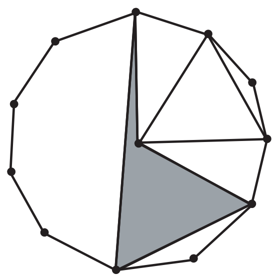


Figure 6: Lower bound

Any line through the center point has at least 5 points on each side, so no edges of the shaded pseudo-triangle are convex hull edges, and each of its three corners has vertex degree at least 4. But in order to complete the triangulation outside the shaded pseudo-triangle, at least one edge must be connected to one of these corners. ∎

5 Pseudo-triangulations of bounded face degree

It is particularly important in this section to recall that the term pseudo-triangulation refers to *minimum* pseudo-triangulations. We describe an iterative construction for a pseudo-triangulation for a set of points in general position. First we form and triangulate the convex hull of the set of points (note that every triangulation is a minimum pseudo-triangulation for a set of points in convex position). Then we insert each internal point in turn into the pseudo-triangulation. To insert a point into a triangle we simply connect two arbitrary vertices of the respective triangle to the new vertex (see Figure 7).

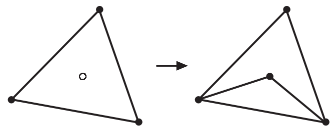


Figure 7: Inserting a vertex into a triangle.

To insert a point into a quadrilateral, we observe that the extensions of the edges that form the side of the pseudo-triangle that consists of two edges partition the pseudo-triangle into three regions (see Figure 8). We choose two of the vertices of the pseudo-triangle depending on which region the points falls into and connect them to the new vertex (see Figure 8).

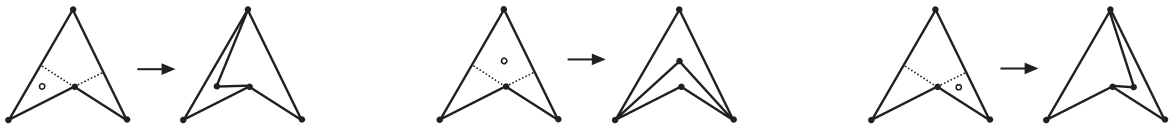


Figure 8: Inserting a vertex into a quadrilateral.

It is not difficult to see how the incremental procedure described above can be implemented, using standard data structures, to run in time $O(n \log n)$ for point sets S of size n. Constructing the convex hull of S, triangulating its interior and building a point location structure for the resulting triangulation $\mathcal{T}$ are all standard procedures that take $O(n)$ time after the set S has been sorted (cf. [5]). It remains to show how to insert the interior points into the incrementally changing pseudo-triangulation in $O(\log n)$ time per point. Since the host triangle in $\mathcal{T}$ can be determined in $O(\log n)$ time by planar point location, it suffices to observe that in each triangle T of $\mathcal{T}$, if the points are inserted in order of increasing x-coordinate, then the pseudo-triangles that partition T, when restricted to points with x-coordinate greater than the most recently inserted point, form a sequence of linearly separable slabs. Thus the associated dynamic point location problem is essentially one-dimensional and can be implemented by straightforward modification of any one of the many optimal dynamic dictionary search structures.

Theorem 8 *For any set S of n points in the plane, a minimum pseudo-triangulation of S, using pseudo-triangles with at most four vertices, can be constructed in $O(n \log n)$ time.*

6 Remarks and open problems

It is natural to ask if the upperbounds on vertex degree and face degree of pseudo-triangulations can be realized simultaneously. Figure 9 illustrates the maximum-vertex-degree-constrained and the maximum-face-degree-constrained pseudo-triangulations (as constructed by the procedures of this paper) for a common point set. The obvious generalization of this example can be used to show a tradeoff between maximum vertex degree and maximum face degree of minimum pseudo-triangulations. Specifically, there exist point sets of size n for which any minimum pseudo-triangulation has a vertex degree and face degree whose product is $\Theta(n)$.

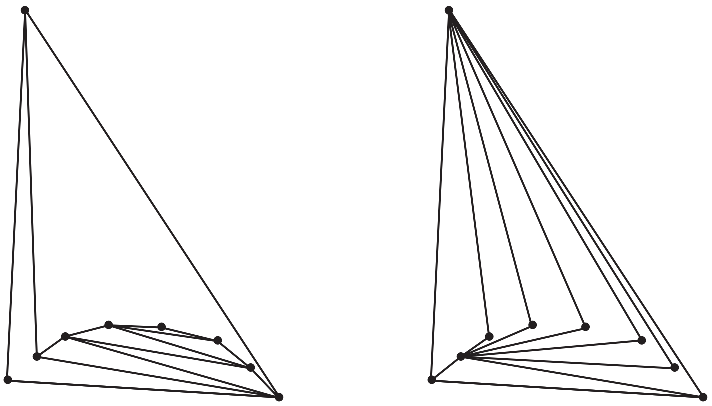


Figure 9: Minimum vertex-degree and minimum face-degree pseudo-triangulations

It remains a open problem to find an efficient algorithm to determine, for a given point set, a minimum pseudo-triangulation of minimum maximum vertex degree. Dynamic and kinetic versions of the bounded degree pseudo-triangulation problem are also of interest.

References

[1] P. K. Agarwal, J. Basch, L. J. Guibas, J. Hershberger, and L. Zhang. Deformable free space tilings for kinetic collision detection. In *Proc. 5th Workshop Algorithmic Found. Robotics*, 2000.

[2] G. S. Brodal and R. Jacob. Dynamic planar convex hull with optimal query time and $O(logn \cdot \log\log n)$ update time. In *Scandinavian Workshop on Algorithm Theory*, pages 57–70, 2000.

[3] T. M. Chan. Dynamic planar convex hull operations in near-logarithmic amortized time. In *Proc. 40th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 92–99, 1999.

[4] B. Chazelle, H. Edelsbrunner, M. Grigni, L. J. Guibas, J. Hershberger, M. Sharir, and J. Snoeyink. Ray shooting in polygons using geodesic triangulations. *Algorithmica*, 12:54–68, 1994.

[5] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications.* Springer-Verlag, Berlin, 1997.

[6] M. Goodrich and R. Tamassia. Dynamic ray shooting and shortest paths in planar subdivision via balanced geodesic triangulations. *J. Algorithms*, 23:51–73, 1997.

[7] J. Hershberger and S. Suri. Applications of a semi-dynamic convex hull algorithm. *BIT*, 32:249–267, 1992.

[8] D. Kirkpatrick, J. Snoeyink, and B. Speckmann. Kinetic collision detection for simple polygons. In *Proc. 16th ACM Sympos. Comp. Geom.*, pages 322–330, 2000.

[9] D. Kirkpatrick and B. Speckmann. Separation sensitive kinetic separation structures for convex polygons. In *Proc. Japan Conference on Discrete and Computational Geometry*, Lecture Notes Comp. Sci. Springer Verlag, 2000. To appear.

[10] M. H. Overmars and J. van Leeuwen. Maintenance of configurations in the plane. *J. Comput. Syst. Sci.*, 23:166–204, 1981.

[11] M. Pocchiola and G. Vegter. Minimal tangent visibility graphs. *Comput. Geom. Theory Appl.*, 6:303–314, 1996.

[12] M. Pocchiola and G. Vegter. Pseudo-triangulations: Theory and applications. In *Proc. 12th ACM Sympos. Comp. Geom.*, pages 291–300, 1996.

[13] M. Pocchiola and G. Vegter. Topologically sweeping visibility complexes via pseudo-triangulations. *Discrete Comp. Geom.*, 16:419–453, 1996.

[14] D. Randall, G. Rote, F. Santos, and J. Snoeyink. Counting triangulations and pseudo-triangulations of wheels. In *Proc. 13th Canad. Conf. Comp. Geom.*, 2001. To appear.

[15] I. Streinu. A combinatorial approach to planar non-colliding robot arm motion planning. In *Proc. 41st FOCS*, pages 443–453, 2000.