Counting and enumerating pseudo-triangulations with the greedy flip algorithm*

Hervé Brönnimann[†] Lutz Kettner[‡] Michel Pocchiola[§] Jack Snoeyink[¶]

Abstract

This paper studies pseudo-triangulations for a given point set in the plane. Pseudo-triangulations have many properties of triangulations, and have more freedom since polygons with more than three vertices are allowed as long as they have exactly inner angles less than π. In particular, there is a natural flip operation on every internal edge. We establish fundamental properties of pseudo-triangulations of point sets. We also present an algorithm to enumerate the pseudo-triangulations of a given point set, based on the greedy flip of Pocchiola and Vegter. Our two independent implementations agree, and allow us to experimentally verify or disprove conjectures on the numbers of pseudo-triangulations and triangulations of a given point set. (For example, we establish that $\#T \leq \#PT$ for all sets of $n \leq 10$ points.)

1 Introduction

Algorithms that perform computations on sets of points in the plane frequently benefit from using the points to decompose the plane into simpler regions: triangulations, Voronoi diagrams, visibility maps, and Delaunay tessellations are good examples [14]. Decompositions called pseudo-triangulations or geodesic triangulations have been studied for convex sets and for simple polygons in the plane because of their applications to visibility [35, 36], ray shooting [13, 18], covering and separation [38], and stretchability of pseudo-lines arrangements [40]. They have been used in a number of kinetic data structures (KDSs) for collision detection among moving objects in the plane [1, 26, 27] because they can be maintained by edge flips and can form a partition of the free space whose size is related to minimum link separators of the objects. Streinu used them in an elegant proof that it is always possible to unfold a chain in the plane without self-intersection [46].

We define pseudo-triangulations of point sets in Section 2.1. Pseudo-triangulations possess versatility and uniformity properties that make them worthy of study. For instance, there is an edge-flip operation that applies to any internal edge in a pseudo-triangulation, unlike the edge-flip operation in triangulations. In Section 2.2, we show how to implement this operation efficiently, and study the graph of pseudo-triangulations, in which two pseudo-triangulations are adjacent if they differ by a single edge flip. We show that this graph is connected, that its diameter is always $O(n^2)$, and that it is the graph of an abstract polytope. It has since been proven [43] that this abstract polytope can in fact be geometrically realized.

We then use edge-flips to enumerate all pseudo-triangulations of a given point set. There have been many interesting results on counting and enumerating triangulations for a given set of points in the plane. There have been a series of upper bounds on the maximum number of triangulations, $\#T(S)$, of a given n-point set S. A count of

*Research by the first author was partially supported by NSF Career Grant #0133599. A preliminary version appeared in *Proc. ALENEX*, 2005.
[†]Polytechnic University, CIS, Six Metrotech, Brooklyn NY 11201; hbr@poly.edu
[‡]Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany; kettner@mpi-sb.mpg.de.
[§]Ecole Normale Supérieure Paris, 45 Rue d'Ulm, 75230 Paris Cedex 05; pocchiola@di.ens.fr.
[¶]Dept Comp. Sci., Sitterson Hall, UNC-Chapel Hill, Chapel Hill, NC 27599-3175 USA; snoeyink@cs.unc.edu.

$\#T(S) \leq 59^{n+o(n)}$ by Santos and Seidel [44] recently replaced the previous best of $\#T(S) \leq 2^{8.12n+O(\log n)}$ by Denny and Sohler [16]. There are examples of point sets with many triangulations that establish a lower bound of $\#T(S) \geq 2^{3n-Theta(\log n)}$ [21]. Aichholzer [3] has a counting algorithm (that can be executed from a web page for small point sets [2]) and Bespamyatnikh [10] and Ray and Seidel [42] present enumeration algorithms. There remain elementary open questions, such as what point sets have the most and the fewest triangulations. (Aichholzer [2] maintains a list of the leading examples for up to 20 points.)

Less is known about the number of pseudo-triangulations, $\#PT(S)$, of a given point set S. Even the following conjecture is open:

Conjecture 1 *[15] For any set S of points in general position in the plane, $\#T(S) \leq \#PT(S)$ with equality iff the points are in convex position.*

Randall et al. [41] have established an upper bound $\#PT(S) \leq 3^i \#T(S)$ for any point set S with i points inside the convex hull. When combined with the bound on the number of triangulations, this gives $\#PT(S) \leq 2^{7.47n+o(n)}$. Bespamyatnikh has extended his enumeration algorithm to pseudo-triangulations, but has yet to implement it. Also, his algorithm cannot take a fixed set K of edges and enumerate only the pseudo-triangulations which contain K.

Our algorithm, presented in Section 3, is based on the greedy flip algorithm of Pocchiola and Vegter for computing the visibility complex of a scene of n convex objects in the plane [36]. As such, our technique can also enumerate the pseudo-triangulations that contain a given set K of edges. In Section 4, we provide some implementation details; we have produced two independent implementations, which may be obtained from `www.cs.poly.edu/pstoolkit/`. and `www.cs.unc.edu/Research/compgeom/pseudoT/` In Section 5 we present the results of experiments that explore basic conjectures on the number of pseudo-triangulations and triangulations. Both implementations agree in these experiments.

Note that Tutte [47] and others have studied the number of topological embeddings of triangulations and rooted triangulations when the locations of vertices are not specified. Li and Nakano [31] enumerate topologically-distinct triangulations with a prescribed number of points on their boundary. We focus strictly on the geometric questions when the vertex set must be a given set of points in the plane.

This work was begun at a Bellairs workshop on pseudo-triangulations organized by Ileana Streinu and partially supported by the NSF. The published results by the participants include the numbers of pseudo-triangulations of special point configurations [41], the existence of pseudo-triangulations with bounded degree [23, 24], and an analysis of the flip graph [43]. Especially this last work quotes some of the results on flipping contained in this paper.

2 Graph of pseudo-triangulations

2.1 Definitions

Pseudo-triangulations were defined by Pocchiola and Vegter [37] for the case of 2-dimensional convex sets. In this paper, we are solely concerned with pseudo-triangulations of point sets. One could replace each point p by a disk with center p and radius ϵ, for some small $\epsilon > 0$, and work within their framework. There are many subtleties involved, however, which warrant a study of the case of point sets in their own right. For instance, two disks have four bitangents, only three of which can be non-intersecting; if we collapse the disks to points, all bitangents collapse to a single edge. To ease the reader's task and prevent circular dependencies, we do not reference the case of convex sets except in the proof of the Flip Property (Theorem 6).

Pseudo-triangles. A *pseudo-triangle* is a simple polygon in the plane that has exactly three vertices, called *corners*, with internal angle less than π. The three corners of a pseudo-triangle decompose its boundary into three concave chains. Let T be a pseudo-triangle. A *tangent* to T is a line l in the plane that goes either through a corner p of T and separates the two edges of T incident upon p, or through a non corner p of T and does not separate the two edges of T incident upon p. A pseudo-triangle has exactly one tangent line parallel to any given line.

Pseudo-triangulations. Let P be a set of n points in general position (i.e., no three collinear points) and let E be the set of $n(n-1)/2$ undirected line segments with endpoints in P (edges for short). For the purpose of this paper, we assume that there are no two parallel edges and no edge parallel to the x-axis.[1]

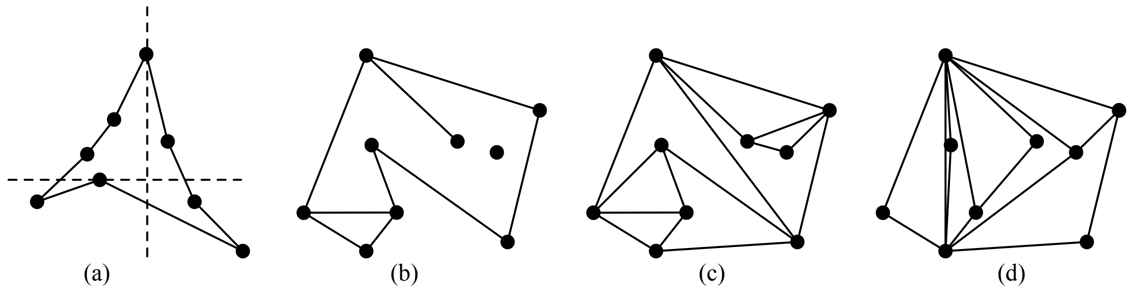

Figure 1: (a) A pseudo-triangle and its horizontal and vertical tangents. (b) An acyclic planar set of edges. (c) A maximal acyclic planar set of edges or, put differently, a pseudo-triangulation. (d) The canonical sorted pseudo-triangulation.

A subset H of E is called *planar* if its edges are pairwise interior disjoint, and is called *acyclic* if for any endpoint p of an edge of H there is a line through p that leaves the edges of H incident upon p all on the same side.

Following Streinu [46] we define a *pseudo-triangulation*[2] of P to be a maximal (for the inclusion relation), acyclic and planar subset of E; note that the set of edges of the convex hull of P is included in every pseudo-triangulation. (See Figure 1 for an illustration.) For completeness, we state and prove the following:

Lemma 2 ([46, Theorem 3.1]) *The bounded faces of the subdivision of the plane induced by a pseudo-triangulation of P are pseudo-triangles. Furthermore the number of pseudo-triangles of the subdivision is $n-2$ and its number of edges is $2n-3$.*

Proof. (Adapted from [37, Lemma 2]) Let R be a planar and acyclic set of edges containing the edges of the convex hull of P. Assume that some bounded face of the induced subdivision is not a pseudo-triangle; from this we shall derive that R is not maximal. This means that this face is not simply connected or that its exterior boundary contains at least 4 corners. In both cases we add an edge to R as follows. Take a minimal length curve homotopy equivalent to the curve formed by the part of the exterior boundary of the face that goes through all corners of the exterior boundary but one. This curve contains an edge not in R and the addition of this edge to R does not violate its acyclicity nor its planarity; hence R is not maximal.

Let R be a pseudo-triangulation and let Q be the set P minus its two points whose y-coordinates are extremal. Since R is acyclic, the map that associates with a pseudo-triangle of the subdivision induced by R the touching point of its horizontal tangent line is one-to-one; furthermore the image of this map is Q since all bounded faces are pseudo-triangles. Thus the number of pseudo-triangles is $n-2$. The last result is then an easy application of Euler's relation for planar graphs. □

For points in convex position, the set of pseudo-triangulations is exactly the set of triangulations; the external angle of each vertex on the convex hull is greater than π, so triangulations are acyclic. We define a canonical *sorted pseudo-triangulation*, as in Figure 1(d), for any set of n points in general position by the following construction: sort the

[1] These two restrictions can be lifted, and indeed should be in a good implementation. In section 5, we apply the algorithm to the point sets from the database of Aichholzer et al. which obey these restrictions.

[2] "Minimum" pseudo-triangulation in her terminology.

points lexicographically by (x, y) coordinates, and form a triangle with the first three points; then for each subsequent point in order, add one pseudo-triangle by creating two tangents to the convex hull. This pseudo-triangulation, called incremental in [1], has been used in collision detection. See Section 3.4 for an algorithm.

2.2 Edge flips in pseudo-triangulations

In a triangulation, an *edge flip* replaces any edge whose adjacent triangles form a convex quadrilateral by the opposite diagonal of that quadrilateral. Edge flips, sometimes known as a Lawson flips, are useful tools to study the properties of triangulations and to generate them algorithmically [21, 29, 33]. An almost identical flip operation is defined for pseudo-triangulations of disks by Pocchiola and Vegter [36].

We now show that edge flips are even nicer in pseudo-triangulations of point sets, because any edge that is inside the convex hull can be flipped.

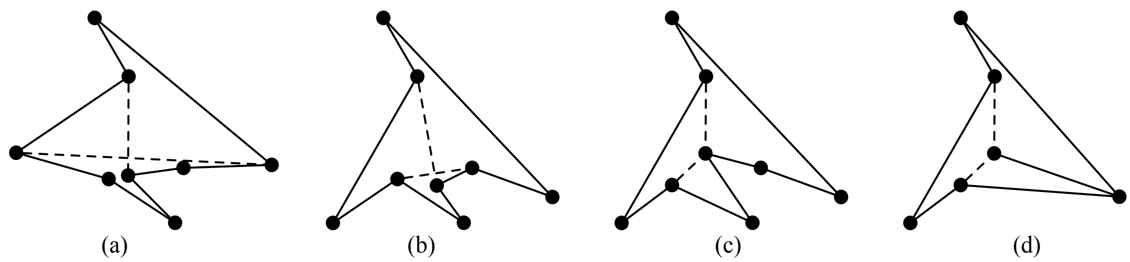

Figure 2: Four pseudo-quadrangles; the last one, (d), uses both sides of one segment. Each pseudo-quadrangle has two diagonals (dashed segments), one on each shortest paths that joins opposite corners. Each diagonal form a pair of pseudo-triangles, and an edge-flip replaces one diagonal with the other.

Consider an edge e that is adjacent to two neighboring pseudo-triangles. Each endpoint of e is a corner in at least one of the neighboring pseudo-triangles, since each vertex has at most one angle that is greater than π, see Figure 2 for examples. We observe that removing edge e merges the two neighboring pseudo-triangles into a “pseudo-quadrangle”: At each endpoint of e either two corners merge into one corner or one corner merges with an angle greater than π. Thus, the six corners of the original two pseudo-triangles become the four corners of a pseudo-quadrangle.

We can define a *diagonal* for a pseudo-quadrangle by connecting opposite corners with a shortest path through the interior. Such a shortest path coincides with parts of the boundary except for exactly one straight edge in the interior, which splits the pseudo-quadrangle into two pseudo-triangles. Since there are two pairs of opposite corners, a pseudo-quadrangle has two diagonals.

Now, any non-hull edge e is one diagonal of a pseudo-quadrangle formed by removing e. We define the *edge-flip* for e as the operation that removes e and replaces it with the other diagonal. Figure 2 shows four examples. From the preceding discussion, we can observe:

Lemma 3 *Any non-hull edge in a pseudo-triangulation can be flipped. The edge-flip operation replaces the non-hull edge and its two incident pseudo-triangles with a new edge and two new pseudo-triangles. Flipping the new edge restores the original pseudo-triangulation.*

2.3 Graph of pseudo-triangulations

We now show that the graph of pseudo-triangulations, in which two pseudo-triangulations are adjacent if they differ by a single edge flip, is connected.

Formally, the *graph of pseudo-triangulations* for a given set of points contains a node for each possible pseudo-triangulation. An arc connects two nodes if the two corresponding pseudo-triangulations differ by a single edge flip. The graph is undirected since flips are reversible.

The *edge-flip distance* between any two pseudo-triangulations is the number of edge flips necessary to change one into the other, i.e., the shortest path between them in the graph. We show that the flip graph is connected and bound its diameter. Rote, Streinu, and Santos have extended this result to a beautiful analysis of the flip graph of pseudo-triangulations, showing that it is polytopal, has a geometric embedding in $\mathbb{R}^{2n-3}$ and relating it to minimally rigid graphs [43].

Lemma 4 *The graph of pseudo-triangulations is connected and the edge-flip distance between any two pseudo-triangulations is $O(n^2)$ for a given set of n points.*

Proof. We show that one can flip from any pseudo-triangulation to the canonical sorted pseudo-triangulation in $O(n^2)$ steps. Since flips are reversible, this is sufficient to establish the lemma.

For a pseudo-triangulation that is different from the sorted pseudo-triangulation, start at the rightmost vertex and flip all incident non-hull edges using less than n flips. In the pseudo-quadrangle formed by removing such a non-hull edge, the rightmost vertex is a corner; the flipped edge in this pseudo-quadrangle cannot attach to the rightmost vertex. Thus, each flip removes an incident non-hull edge from the rightmost vertex. What remains is a rightmost vertex with two hull edges. The corresponding pseudo triangle forms a convex chain opposite to the rightmost vertex, which is the same configuration as in the sorted pseudo-triangulation. We drop the rightmost vertex and continue with the pseudo-triangulation on the remaining $n-1$ points recursively. This procedure performs a total of $O(n^2)$ flips. □

Since pseudo-triangulations of points in convex position are identical to triangulations, the lower bound constructions for $\Omega(n^2)$ flipping distance for triangulations with points in convex position apply for pseudo-triangulations [45].

The graph of pseudo-triangulations has an even stronger connectivity property. If we consider only the pseudo-triangulations that contain a chosen set of edges $K \subseteq E$, then we get a subgraph induced by the nodes corresponding to those pseudo-triangulations, whose arcs join nodes that correspond to flips of the edges not in K. This subgraph is connected. As we will see, this is a simple consequence of our enumeration algorithm.

Let X be the set of acyclic and planar subsets of E that contain the set of hull-edges, ordered by reverse inclusion and augmented with a minimum element (it already has a maximum element, namely $\emptyset$). Therefore X is a lattice, and the edge-flip operation provides the diamond property of abstract polytopes. (See e.g. the article by McMullen [32] or the appendix of [36] for the notions related to abstract polytope.) As a consequence of the last observation and of the strong connectivity of X mentioned in the last paragraph, we have:

Lemma 5 *The lattice X is an abstract polytope of dimension $2n-3-h$, where h is the number of edges on the convex hull. Its set of vertices is the set of pseudo-triangulations of X. Its 1-skeleton is the flip graph of pseudo-triangulations of S. This abstract polytope is simple.*

For further developments on the polytope X and especially for a geometric realization of X, see the paper [43].

3 Enumerating pseudo-triangulations

Our goal is to enumerate the set of pseudo-triangulations over a given set of points. To this end we are going to define a total order $\prec$ on the set of edges and a binary tree of pseudo-triangulations whose leaves considered as increasing sequences of edges are the pseudo-triangulations ordered lexicographically; furthermore two adjacent pseudo-triangulations in the tree are either identical or related by a flip operation. Our enumeration algorithm is a traversal of this tree guided by the aforementioned total order $\prec$. Our technique can also be applied to enumerate the pseudo-triangulations that contain a given set of edges.

3.1 The flip property

We introduce some definitions and prove a flip property that is essential to prove the correctness of the enumeration algorithm. For each edge $e \in E$, define $\Theta(e)$ as the angle in $[0, \pi)$ that the edge e directed upward makes with the positive horizontal direction Ox.

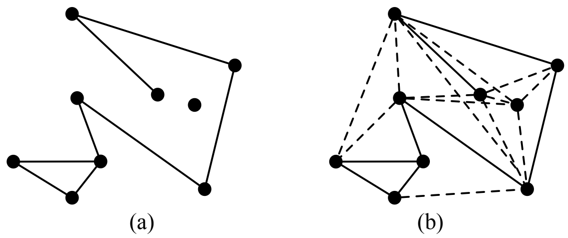

Figure 3: (a) An acyclic and planar subset K of E, and (b) the set $E_K = \{e \mid K \cup \{e\} \text{ is acyclic and planar}\}$.

Given an acyclic planar subset of edges $K \subseteq E$, we denote by E_K the set of edges $e \in E$ that can be used to complete K to a pseudo-triangulation, i.e., such that $K \cup \{e\}$ is acyclic and planar. See Figure 3 for an illustration.

We define a partial order $\prec_K$ on E_K as follows: $e \prec_K e'$ if there exists a sequence of edges $e_1 = e, e_2, \ldots, e_k = e'$ such that e_i and e_{i+1} share a common endpoint and angles $\Theta(e_i) < \Theta(e_{i+1})$. According to the general position assumption, two edges sharing an endpoint have different angles and therefore are comparable. It follows that the edges of a pseudo-triangle are pairwise comparable and are encountered in increasing order when traversing the boundary of the pseudo-triangle counterclockwise, starting from its point of horizontal tangency. Following [36, Lemma 7] we observe that two crossing edges in E_K are the diagonals of some pseudo-quadrangle (with edges in E_K) and consequently they are comparable with respect to $\prec_K$.

A *filter* for a poset $(X, \prec)$ is a subset I of elements such that for any $x \prec y$, if $x \in I$ then $y \in I$. In particular, to each angle θ corresponds a filter $I_{K,\theta}$ of the poset $(E_K, \prec_K)$ whose elements are the edges e' of E_K whose angle $\Theta(e')$ is greater than θ. Note that $I_{K,0} = E_K$, and $I_{\emptyset,0} = E$.

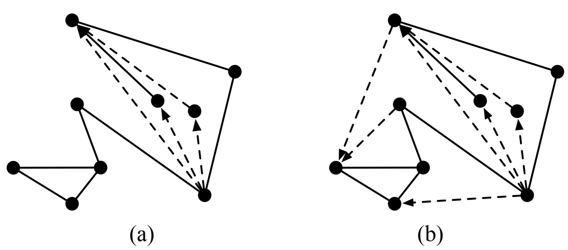

Figure 4: The set $G(I_{\pi/2})$ is constructed by adding edges from the filter $I_{\pi/2}$ — i.e., edges with increasing angles greater than $\pi/2$ — until a pseudo-triangulation is formed. (a) illustrates that it is not sufficient to consider angles in $[\pi/2, \pi)$ only, but by "wrapping around" (b) we do complete a pseudo-triangulation.

For any filter I of the poset $(E_K, \prec_K)$, we define a maximal planar acyclic set of edges $G(I) = \{e_1, e_2, \ldots, e_k\}$ recursively: edge e_1 is minimal in I, and, for $i \geq 1$, edge e_{i+1} is minimal in the set of edges $e \in I \setminus \{e_1, \ldots, e_i\}$ such that $K \cup \{e_1, \ldots, e_i, e\}$ is acyclic and planar. Since two edges that cross or that share an endpoint are comparable, the set $G(I)$ is well-defined, up to the choice of the minimal element at each step.

We would like for $G(I)$ to be a pseudo-triangulation, and for this it suffices to make sure that $G(I)$ has $2n-3$ edges. This is not always possible, however, due to the fact that we chose the principal determination of Θ in $[0, \pi)$, therefore introducing a discontinuity in the comparisons, and forcing the algorithm to stop perhaps too early. (See Figure 4.) To circumvent this, we replace, in the previous definitions, the set of edges E_K by its infinite cover $\mathbb{E}_K = E_K \times \mathbb{Z}$: elements of $\mathbb{E}_K$ are still called edges and the angle $\Theta(v)$ of an edge $v = (e, k)$ of $\mathbb{E}_K$ is defined to be the real $\Theta(e) + k\pi$. The operator on $\mathbb{E}_K$ that increases the angle of an edge by π is denoted ι. It is not hard to see that if I is a filter of $(\mathbb{E}_K, \prec_K)$ then its projection on the first factor E_K is onto, from which we deduce that $G(I)$ is a pseudo-triangulation. In this context we redefine the flip operation as follows: to flip e in $G(I)$ means to replace e by $\iota(e)$ if $e \in \mathbb{K} := K \times \mathbb{Z}$ or if e is a hull-edge, otherwise to perform an edge-flip on e and assign the angle of the diagonal by adding a multiple of π to fall into the range $(\Theta(e), \Theta(e) + \pi)$.

The pseudo-triangulation $G(I_{\emptyset,0})$ is called the *horizontal greedy pseudo-triangulation* and plays a particular role in our enumeration algorithm. Further below, we explain how to efficiently compute this pseudo-triangulation.

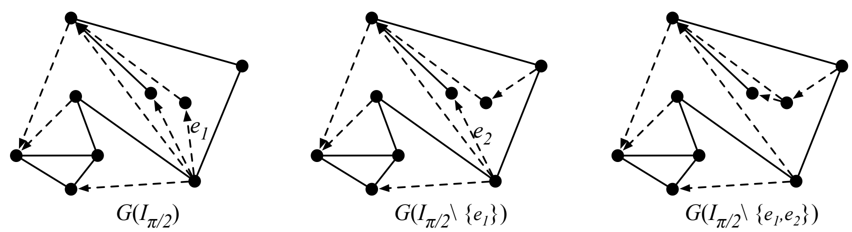

Figure 5: Illustration of the Flip Property for Points.

We are now in a position to state the *flip property*. This property states that flipping a minimal edge in a pseudo-triangulation of the form $G(I)$ results in a pseudo-triangulation of the form $G(J)$, and this will be crucial in our enumeration algorithm. See Figure 5 for an illustration.

Theorem 6 (Flip Property for Points) *Let I be a filter of the poset $(\mathbb{E}_K, \prec_K)$ and let e be minimal in I. Then $G(I \setminus e)$ is obtained from $G(I)$ by flipping e.*

The proof relies on the theory of pseudo-triangulations developed for bounded 2-dimensional convex sets in [36]. The heart of the proof is to replace each point $p \in P$ by the disk with center p and radius ϵ, for some small $\epsilon > 0$ and to derive the flip property for points from the "flip property for disks" (cf. [36, Theorem 12] and [7, Theorem 5]). There are many subtleties involved, however. For one things, disks have four bitangents, only three of which can be non-intersecting, and they all map to a single edge of the point set. Nevertheless, with a bit of care, it is possible to carry the flip property from the case of disks to the case of points.

We briefly recall the terminology and the results of [36] needed to our purpose. Let $O_1, O_2, \ldots, O_n$ be a collection of n pairwise disjoint bounded closed convex subsets of the plane with nonempty interiors and regular boundaries (obstacles or disks for short). We assume that there is no line tangent to three disks. A *bitangent* is a closed undirected line segment whose supporting line is tangent to two disks at its endpoints. A *free* bitangent is a bitangent whose interior lies in free space the complement of the disks. In the following considerations all bitangents are free. A set of (free) bitangents is called *planar* if its elements are pairwise disjoint. A *pseudo-triangulation* (of the O_is) is a maximal — for the inclusion relation — planar set of bitangents. It is known that a pseudo-triangulation contains $3n-3$ bitangents that decomposes the convex hull of the disks into $2n-2$ *pseudo-triangles* where in this context a pseudo-triangle is a simply connected region of the plane whose boundary consists of three convex curves that share a tangent at their common endpoint and which is included in the triangle formed by the three endpoints of these convex curves.

We denote by B the set of (free) bitangents and we introduce its infinite cover $\mathbb{B} = B \times \mathbb{Z}$: elements of $\mathbb{B}$ are still called bitangents; the angle $\Theta(v)$ of the bitangent $v = (b, k) \in \mathbb{B}$ is defined to be the real $\Theta(b) + k\pi$ where $\Theta(b)$ is the angle in $[0, \pi)$ that b oriented upward makes with the horizontal positive direction Ox, and the direction of $v \in \mathbb{B}$ is the unit vector $(\cos\Theta(v), \sin\Theta(v)) \in \mathbb{S}^1$; the operator that increases the angle of a bitangent by π is denoted ι.

Given a planar subset H of B we introduce the set B_H of bitangents of B that cross properly no bitangent of H (thus $H \subseteq B_H$). The set $\mathbb{B}_H$ is endowed with a partial order $\prec_H$ defined as follows: $b \prec_H b'$ if there exists a sequence of bitangents $b_1 = b, b_2, \ldots, b_k = b'$ in $\mathbb{B}_H$ such that b_i and b_{i+1} touch the same oriented disk[3] and $\Theta(b_i) < \Theta(b_{i+1})$ Each (proper) filter I of $(\mathbb{B}_H, \prec_H)$ is associated with its so-called greedy pseudo-triangulation

$$G(I) = \{b_1, b_2, \ldots, b_{3n-3}\} \subset I$$

defined as follows: (1) b_1 is minimal in I, and (2) b_{i+1} is minimal in the subset of bitangents of I crossing none of the bitangents $b_1, b_2, \ldots, b_i$. Since crossing bitangents are comparable (cf. Lemma [36, Lemma 7]), $G(I)$ is well-defined and is a superset of the set of minimal elements of I. To flip b in $G(I)$ means to replace b by $\iota(b)$ if $b \in \mathbb{H} := H \times \mathbb{Z}$ or if b is a hull bitangent, otherwise to replace b by the second diagonal (with the appropriated angle) of the pseudo-quadrangle obtained by merging the two pseudo-triangles incident upon b in $G(I)$.

Theorem 7 (Flip Property for Disks [7, 36]) *Let b be minimal in the filter I of the poset $(\mathbb{B}_H, \prec_H)$. Then $G(I \setminus b)$ is obtained from $G(I)$ by flipping b.*

According to the Flip Property the mapping that associates with the bitangent $b \in \mathbb{B}_H$ the bitangent $b' \in \mathbb{B}_H$ defined by $\{b'\} = G(I \setminus b) \setminus G(I)$ where b is minimal in I is well-defined (because independent of I), one-to-one, and onto; the bitangent b' is denoted $\phi(b; H)$.

We turn now to the proof of the Flip Property for points. We split the proof into several lemmas. The key idea of the proof is to define an epimorphism of posets to carry the Flip Property from the case of disks to the case of points. Before defining this epimorphism we reformulate the greedy procedure in terms more suitable for our subsequent analysis.

Lemma 8 *Let I be a filter of $\mathbb{B}_H$ and let F be* initial *in I, i.e., $I \setminus F$ is a filter. Then $G(I) = G(I(F))$ where $I(F)$ is the filter of the poset $\mathbb{B}_{H \cup G(F)}$ defined by $I(F) = I \cap \mathbb{B}_{H \cup G(F)}$. A similar result holds when taking $K, \mathbb{E}_K$ for H and $\mathbb{B}_H$.*

Proof. Since F is initial $G(F)$ is a subset of $G(I)$ from which the lemma follows easily.□

Now we turn to the construction of the epimorphism. For $\epsilon > 0$ let $O_i(\epsilon)$ be the disk with center p_i and radius ϵ. Since the points are in general position there exists a real $\epsilon_0 > 0$ such that for all $\epsilon < \epsilon_0$ no line pierces three of the disks $O_i(\epsilon)$ and no horizontal line pierces two disks. (Remember that we assumed that no two points have the same ordinate.) We choose such an ϵ and we introduce the set B of $2n(n-1)$ (free) bitangents of the $O_i(\epsilon)$s and we denote by η the four-to-one mapping that associates with a bitangent b in B tangent to the disks O_i and O_j the edge $\eta(b)$ of E with endpoints p_i and p_j. Our first lemma provides a characterization of acyclicity in terms of the crossing predicate.

Lemma 9 *Let K be a planar subset of E. Then K is acyclic iff for all maximal (for the inclusion relation) planar subset H of $\eta^{-1}(K)$ one has $\eta(H) = K$.*

[3] An oriented disk is a disk with a "direction", "sense", or "orientation" assigned to its boundary. An oriented disk and a bitangent touch each other or are tangent to each other if their directions at the point of touching are the same.

Proof. The 'if part' is easy. To prove the 'only if part' we show that if there exists an edge $e \in K$ and a maximal planar subset H of $\eta^{-1}(K)$ such that $e \notin \eta(H)$ then K is not acyclic. Let p and q be the endpoints of e and let p^+ (resp. p^-) be the set of edges $e' \in K$ such that (1-) p is endpoint of e', i.e, $e' = [p, r]$ for some point r, and (2-) the triplet of points p, r, q is oriented counterclockwise (resp. clockwise).

The assumption that $e \notin \eta(H)$ is equivalent to say that for all $b \in \eta^{-1}(e)$ there exists a bitangent b' of H such that b and b' are crossing. A simple case analysis shows that this is equivalent to say that the sets $p^+ \cup q^+$, $p^+ \cup q^-$, $p^- \cup q^+$ and $p^- \cup q^-$ are nonempty; from which we deduce that p^+ and p^- (or q^+ and q^-) are nonempty and consequently that K is not acyclic. $\square$

Our next lemma is the key to the construction of an epimorphism from some $B_H \subseteq \eta^{-1}(E_K)$ onto E_K.

Lemma 10 *Let K be an acyclic and planar subset of E and let H be a maximal (for the inclusion relation) planar subset of $\eta^{-1}(K)$. Then*
1. *$\eta(\mathbb{B}_H) = \mathbb{E}_K$, and*
2. *if $b \prec_H b'$ then $\eta(b) \preceq_K \eta(b')$.*

Proof. Claim (1) is consequence of Lemma 9: indeed let $K' = K \cup \{e\}$ be planar and acyclic and let $H' \supseteq H$ be a maximal planar subset of $\eta^{-1}(K')$ that contains H. According to Lemma 9 applied to the pair K', H' one has $\eta(H') = K'$ and consequently some bitangent of $\eta^{-1}(e) \in B_H$. Claim (2) is a simple consequence of Claim(1) and the observation that if b and b' touch the same oriented disk with $\Theta(b) < \Theta(b')$ then $\eta(b)$ and $\eta(b')$ share a common endpoint and $\Theta(\eta(b)) \leq \Theta(\eta(b'))$ with equality iff $\eta(b) = \eta(b')$. $\square$

In other words, the restriction η_H of η to $\mathbb{B}_H$ is a mapping onto $\mathbb{E}_K$ that preserves the orderings. We show that η_H preserves also the greedy pseudo-triangulations.

Lemma 11 *Let K be an acyclic and planar subset of E, let H be a maximal planar subset of $\eta^{-1}(K)$, and let I be a filter of $(\mathbb{E}_K, \prec_K)$. Then*
1. *$\eta_H^{-1}(I)$ is a filter of $(\mathbb{B}_H, \prec_H)$*
2. *$\eta(G(\eta_H^{-1}(I))) = G(I)$.*

Proof. Claim (1) is simple consequence of Lemma 10. Claim (2) is proved by induction on the set of planar acyclic subsets K of E ordered by reverse inclusion. Let $J = \eta_H^{-1}(I)$. This is clearly valid if K is maximal since in that case $\mathbb{E}_K = \mathbb{K}$, $\mathbb{B}_H = \mathbb{H}$ and consequently $G(I) = I \setminus \iota(I)$ and $G(J) = J \setminus \iota(J)$ from which we deduce that $\eta(G(J)) = G(I)$. Assume now that K is not maximal and let e be minimal in I. If $e \notin K$ we set $K' = K \cup \{e\}$ and $H' = H \cup G(\eta_H^{-1}(\{e\})$. According to Lemma 8, $G(I) = G(I')$ and $G(J) = G(J')$ where $I' = I(\eta_H^{-1}\{e\})$ and $J' = J(\{e\})$. One can check that $I' = \eta_{H'}^{-1}(J')$ from which the result follows by induction since $K \subset K'$. In case $e \in K$ we replace e by an initial segment of J that contains exactly one element not in K and proceed similarly. $\square$

We are now ready to carry the Flip Property from the case of disks to the case of points.

Proof of Theorem 6. Let $J = \eta_H^{-1}(I)$ and let $F = \eta_H^{-1}(e)$. Note that F is initial in J since $J \setminus F = \eta_H^{-1}(I \setminus e)$ (cf. Lemma 11, Claim 1). Thanks to the Flip Property for disks the set $G(J \setminus F)$ is obtained from $G(I)$ by flipping successively the bitangents of F. Therefore one has

$$G(J \setminus F) = (G(J) \setminus G(F)) \cup \phi(F; H) \setminus F$$

and consequently, according to Lemma 11,

$$G(I \setminus e) = (G(I) \setminus e) \cup \eta(\phi(F, H) \setminus F).$$

Since $G(I \setminus e)$ and $G(I)$ have the same cardinality, namely $2n - 3$, it follows that $G(I \setminus e) \setminus G(I)$ is reduced to a single edge, which is one of the edges of $\eta(\phi(F; H) \setminus F)$. $\square$

3.2 Algorithms for edge flip

In this section we sketch two implementations of edge flip, assuming that the pseudo-triangulation is stored in a data structure that allows us to access its edges in order around a given face. Standard structures for planar subdivisions, such as doubly-connected edge lists or quadedge [14, 19], provide this.

Rotational sweep for edge-flip. We can determine the new diagonal obtained by flipping edge e using a rotational sweep somewhat similar to the rotating caliper [14]. The algorithm proceeds by rotating parallel tangents simultaneously along the interiors of the two pseudo-triangles adjacent to e. Starting from the edge e that we want to flip, the two tangents initially coincide but have opposite orientations. If we sweep through the angles, the two tangents immediately separate and meet again only when they reach the new diagonal. We can discretize this sweep because the tangents rotate around vertices until they are collinear with the next halfedge of a pseudo-triangle. Then they advance to the next vertex. At corners the tangent changes its orientation with respect to the halfedge orientation. The sweep terminates when the tangents again coincide. See Figure 8 below for an implementation.

A matroidal flip algorithm. The predicate in the rotational sweep is a test ordering two vectors. One can also give an implementation that uses only the orientation predicate `left_turn(`p, q, r`)`, which returns true iff the point sequence p, q, r forms a left turn. We can call such an algorithm "matroidal," in that it only uses information about the order type of the points [11, 28]. Such algorithms are usually better in that they have fewer degenerate configurations, lower arithmetic complexity, and generalize to other matroids.

The idea behind the algorithm is to identify the flip as the only diagonal edge on the shortest path connecting the opposite corners. The funnel algorithm of Lee and Preparata can be modified [20, 30] to compute shortest paths in linear time and return the unique edge not on the boundary of the pseudo-quadrangle. Alternatively, one can compute common tangents for the pairs of chains in a pseudo-quadrangle to identify the diagonals. Tangents for two separated chains can be found in $O(\log n)$ time [25, 34]. When computing the visibility graph of a set of convex obstacles, Angelier and Pocchiola [7] use a clever amortization scheme to compute tangents in $O(1)$ time apiece.

3.3 The enumeration algorithm

In this section, we consider the total order $<$ on E and $\mathbb{E}$ induced by Θ. This order is compatible with, and linearly extends, $(E, \prec)$. Although we assume general position, the case of parallel edges could be handled by considering a linear extension of $(E, \prec)$.

In the algorithm, we speak of edges in E as colored red or blue. The red edges are fixed and will not be flipped; the blue edges can be flipped. We now describe the following binary tree $\mathcal{T} = \mathcal{T}(P)$ of {red,blue}-colored pseudo-triangulations of P. Each node of the tree corresponds to a colored pseudo-triangulation G, and we identify the node of the tree with its pseudo-triangulation G. The tree is defined as follows:

1. The root of $\mathcal{T}$ is the horizontal greedy pseudo-triangulation $G(I_{\emptyset,0})$; all its edges are blue.
2. Let G be node of $\mathcal{T}$: If either (i) a blue edge of G has an angle $\geq \pi$ or (ii) all the edges of G are red, then G is a leaf of the tree. Otherwise, let e be a minimal blue edge, e.g., the blue edge with minimum angle $\Theta(e)$. The right child of G is obtained from G by flipping e and its left child is obtained by changing the color of e to red.

A leaf G satisfying 2.(i) is called a blue leaf, and a leaf satisfying 2.(ii) is called a red leaf. Blue leaves stop the algorithm from enumerating a pseudo-triangulation G several times; without stopping for blue leaves, the tree would be infinite, and each pseudo-triangulation would be reported for every value of Θ with the same remainder modulo π.

The algorithm simply explores the tree $\mathcal{T}$ by a depth-first traversal, visiting the left child before the right child, and reporting the red leaves in the order in which they are discovered.

The algorithm is fully described once we explain how to find the minimal blue edge e. In the most direct implementation, the blue edges are stored in a priority queue, ordered by Θ. The edge e at the top of the queue is removed upon descending to either child, and edge e' is enqueued when descending to a right child iff its angle $\Theta(e') < \pi$ (otherwise the right child is a blue leaf and the recursion stops).

It is not necessary to store the priority queue in the recursion stack if we simply add edge e back to the queue when returning to the parent after visiting the right subtree. Thus, the stack grows by $O(1)$ at each recursive call.

Theorem 12 *The set of red leaves of $\mathcal{T}(P)$ ordered from left to right (in the order they are reported by the algorithm) is the set of pseudo-triangulations of P ordered lexicographically by Θ.*

Proof. Let G be a pseudo-triangulation with edges in E and let $G_0, G_1, \ldots G_i, \ldots G_k$ be the path in the tree defined inductively: G_0 is the root the tree and G_{i+1} is the left child of G_i if the minimal blue edge of G_i belongs to G otherwise G_{i+1} is its right child. Let K_i be the set of red edges of G_i, edge e_i be the minimal blue edge of G_i, and filter $I_i = I_{K_i,\Theta(e_i)}$ of E_{K_i}. We claim that

1. $K_i \subseteq G$,
2. $G \setminus K_i \subset I_i$,
3. $G(I_i) = (G_i \setminus K_i) \cup \iota(K_i)$.

from which we deduce that G_k is a red leaf and $G = G_k$.

Claims (1), (2) and (3) are easily proven by induction on i using the Flip Property of the previous section. The proof is finished by noting that the red leaves of the tree are pseudo-triangulations with edges in E. $\square$

Note that the theorem is also valid for any total order $<$ that is a linear extension of $\prec$, yielding a well defined tree $\mathcal{T}_<(P)$. Since Θ induces such a total order and is easy to compute, thanks to the geometry, it is convenient to use it. See Remark 1 below.

Theorem 13 *The height of the tree $\mathcal{T}(P)$ is at most $n(n-1)/2$.*

Remarks. **1.** In this formulation, the algorithm depends on the order Θ of the edges, which is not implied by the orientation of all triplets of points. For this reason, the algorithm as described here is not matroidal. It is, however, possible to give a matroidal algorithm, by selecting for e *any* blue edge that is minimal for the partial order $\prec$. In this case the tree $\mathcal{T}(P)$ isn't uniquely defined, and finding such an edge is more difficult, necessitating the maintenance of the antichain ($\hat{I}$ in the notation of [7]) associated with the current filter I while traversing the tree.

2. Maintaining the dual pseudo-triangulation $G_* = \phi_*(G)$ (in the notation of [7]) while traversing the tree, where $\phi_* = \phi^{-1}$ and where e and $\phi(e)$ have the same color, allows to retrieve e when coming back from the right subtree. Hence instead of storing a recursive stack to remember e on the way up the tree, the algorithm can maintain only G and its dual. This changes the space complexity of the algorithm from quadratic to linear.

3. If K is an acyclic and planar set of edges, then by coloring the edges of K red, and replacing the root of the tree by the horizontal greedy pseudo-triangulation $G(I_{K,0}) \subseteq E_K$, the algorithm enumerates the set of pseudo-triangulations that contain K. Observe that this proves that the graph of pseudo-triangulations is strongly connected.

3.4 The horizontal greedy triangulation

We now explain how to compute $G(I_{\emptyset,0})$. In fact, the algorithm can be adapted by a simple rotation to compute $G(I_\theta)$ for any $\theta \in [0,\pi)$. It is convenient for the exposition (and for the algorithm) to order the points of P by lexicographical order, i.e., $p_1 <_{yx} p_2 <_{yx} \cdots p_n$.

The construction uses *lower* and *upper horizon trees*, defined here as follows. For all point p_i with $1 \leq i < n$, denote by $\ell(p)$ the point p_j, for $j > i$, which minimizes the angle $\Theta(\vec{pp_j}) \in [0,\pi)$. Define $\ell(p_n) = p_n$. Since $\ell^n(p) = p_n$, the set of edges of the form $p\ell(p)$ is a tree whose root is p_n (it is connected because every p_i has a path to p_n, and it has n vertices and $n-1$ edges). We call that tree the *lower horizon tree* and denote it by $T_\ell(P)$.

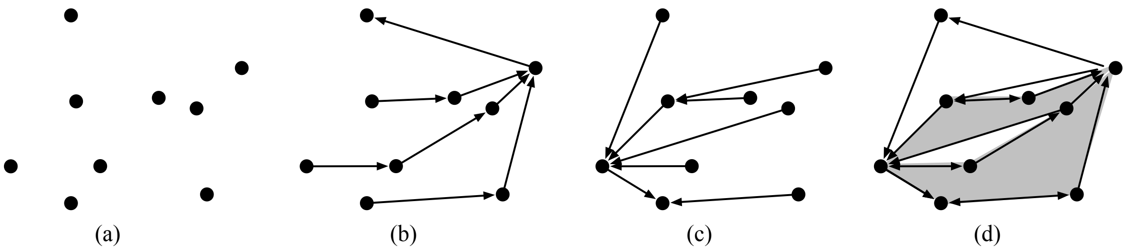

Figure 6: (a) A point set, (b) its lower and (c) upper horizon trees; (d) the superimposition of the horizon trees yields pseudo-quadrangles (shaded) and pseudo-triangles.

COMPUTELOWERHORIZONTREE(P)
Effects: computes $\ell(p_i)$ for every $p_i \in P$

```
ℓ(p_n) ← p_n
for i ← n − 1 downto 1 do
    j ← i + 1
    while right_turn(p_i, p_j, ℓ(p_j)) do
        j ← j + 1
    ℓ(p_i) ← p_j
```

COMPUTEUPPERHORIZONTREE(P)
Effects: computes $u(p_i)$ for every $p_i \in P$

```
u(p_1) ← p_1
for i ← 2 to n do
    j ← i − 1
    while right_turn(p_i, p_j, u(p_j)) do
        j ← j − 1
    u(p_i) ← p_j
```

Figure 7: Computing the lower and upper horizon trees.

Likewise, for point p_i with $1 < i \leq n$, denote by $u(p)$ the point p_j, $j < i$, which minimizes the angle $\Theta(\vec{p}p_j) \in [\pi, 2\pi)$ (define $u(p_1) = p_1$). The set of edges of the form $pu(p)$ is also a tree, of root p_1, which we call the *upper horizon tree* and denote by $T_u(P)$.

The following lemma, first observed in [39], forms the basis of the algorithm. See Figure 6 for an illustration.

Lemma 14 *Let $K = T_\ell(P) \cup T_u(P)$ be the set of edges belonging to the horizon trees.*
(1) K contains all the edges of the convex hull of P.
(2) K decomposes the convex hull of P into regions, each of which is either a pseudo-triangle or a pseudo-quadrangle.
(3) $K \subseteq G(I_{\emptyset,0})$.

With this lemma, the algorithm is straightforward. The pseudo-code is given in Figure 7. It computes $\ell(p)$ for each $p \in P$ by Andrew's variant of Graham's convex hull algorithm [6]. We need the predicate `right_turn`(p, q, r) which returns true if the point sequence p, q, r forms a right turn. (In particular, the inner while loop will stop at $j = n$ since `right_turn`(p_i, p_n, p_n) is always false.)

Note that the algorithm still produces the correct tree if two edges are parallel or three points are collinear, or even if two points have the same ordinate (thanks to the lexicographical order).

Computing $u(p)$ is performed by a similar algorithm. After the initial sorting in $O(n \log n)$ time, both algorithms take $O(n)$ time. Once the horizon trees have been computed, the subdivision can be constructed in linear time, and each region visited to determine if it is a pseudo-triangle or pseudo-quadrangle. A pseudo-quadrangle can be split in time linear to its number of edges, by computing its two diagonals and inserting the one with smaller Θ. Thus, once the points are sorted lexicographically, the algorithm computes $G(I_{\emptyset,0})$ in linear time.

3.5 Complexity analysis

The algorithm spends $O(n)$ time for a flip or a priority queue operation in the worst case, hence time $O(n)$ per edge of the tree. Since the number of edges is the same as the number of internal nodes, which is also half the number of leaves, the algorithm spends amortized time $O(n)$ per pseudo-triangulation.

Using a heap for the priority queue reduces the cost of the priority queue operations to $O(\log n)$. Moreover, using binary search can reduce the complexity of the flip algorithm to $O(\log n)$ as well, at the cost of maintaining the corners of pseudo-triangles (which can be done in $O(1)$ time after a flip) and maintaining the boundary of the pseudo-triangles as splittable queues as in [36].

Unfortunately, this is the time spent per leaf, counting both the n_r red *and* the n_b blue leaves. The following ratio is therefore important for the analyzing the complexity of the algorithm: $\rho = (n_b + n_r)/n_r$. We initially conjectured a bound of 2 on this ratio, which was disproved by experiments (see next section). The currently best upper bound we have is the number of edges of a pseudo-triangulation not on the convex hull, i.e., $2n - 3 - h$.

To conclude, the algorithm is set up in time $O(n \log n)$ to compute the horizontal greedy triangulation $G(I_{\emptyset,0})$ and insert its edges in the priority queue. The running time of the algorithm per red leaf of the tree (i.e., pseudo-triangulation of P) is upper-bounded by $O(\rho n) = O(n^2)$, and can be lowered with more complicated algorithmic machinery to $O(\rho \log n) = O(n \log n)$. All of this is in the worst case.

Note that the average complexity of a pseudo-triangle is $O(1)$, thus on the average the flip will be performed in constant time. We expect that ρ is much smaller than n, although not constant ($\rho = O(\log n)$ seems a tempting conjecture, but we do not have a shred of evidence in support). Thus in practice, we expect that the amortized cost per pseudo-triangulation is much lower than $O(n \log n)$, perhaps $O(\rho)$. In order to state such a result, however, we lack an amortized bound for the flip and an upper bound for ρ.

Note finally that the number of pseudo-triangulations grows exponentially fast, thus limiting the domain of practicality of our algorithm in the low tens (twenties). Thus all of the asymptotic complexities should be taken with a grain of salt. A good implementation will settle for low-complexity algorithms as well as simplicity of the code.

4 Implementation issues

Two independent implementations based on the above algorithm have been developed in order to ensure the correctness of the experimental validation of the conjecture.

4.1 Halfedge data structure

Both implementations chose to represent pseudo-triangulation by a halfedge data structure, a.k.a. doubly connected edge list or DCEL. One implementation is based on CGAL, and described in [17, 22], and the other on an independent halfedge data structure described in [12].

A halfedge data structure (HDS) is an edge-based data structure capable of storing a pseudo-triangulation, or more generally any connected planar set of edges. Each edge is split into two halfedges with opposite orientations. By convention, the halfedges incident to a face are oriented counter-clockwise around the face. An `opposite` pointer links a halfedge to its opposite halfedge, and `next` and `prev` pointer links it to the next halfedge in counterclockwise orientation along the incident face. The incident vertices of a halfedge are named the `source` and `target`, as in [12].

4.2 Flip algorithm

Since the algorithm needs to examine the flip edge and decide whether to actually perform the flip or not, it is advantageous to implement the rotational sweep method. Moreover, in this case, neither the flip algorithm nor the enumeration algorithm need the reverse `prev` pointers, thus saving space and execution time. For simplicity, we present below an implementation that uses `prev` pointers, namely in the `is_corner` function. Eliminating `prev` pointers is possible (see the implementation in `http://geometry.poly.edu/pstoolkit/`) but complicates the pseudo-code.

FINDPSEUDOFLIP(h)
Returns: a pair (h', g') such that the edge joining source(h') and source(g') is the flip of the edge supporting h.

```
g ← opposite(h)
reverse_h←is_corner(h), reverse_g←is_corner(g)
while true do
  {decide which of g or h is the next tangent to jump to the next vertex}
  if rotate_ccw_less(source(h), target(h), source(g)), target(g))
                is the same as (reverse_h≠reverse_g) then
    {test if advancing g crosses over h and thus is the solution}
    if left_turn(source(h), source(g), target(g))≠reverse_g then
      return (h, g)
    {not a solution yet, advance g}
    g ← next(g)
    if is_corner(g) then
      reverse_g←negate(reverse_g)
  else
    {test if advancing h crosses over g and thus is the solution}
    if left_turn(source(h), source(g), target(g))≠reverse_h then
      return (h, g)
    {not a solution yet, advance h}
    g ← next(h)
    if is_corner(h) then
      reverse_h←negate(reverse_h)
```

Figure 8: An implementation of the rotational flip method.

The function FINDPSEUDOFLIP returns two halfedge handles whose source vertices form the endpoints of the flipped diagonal rotated counterclockwise from the old diagonal. The function does not actually flip the diagonal. Note that the result could include the old diagonal, which needs to be considered before removing the old diagonal. In the function, h and g are the halfedges whose source vertices are in contact with the two rotating tangents. The two flags reverse_h and reverse_g indicate the relative orientation of the tangents to the halfedge h and g respectively.

The pseudo-code is presented in Figure 8. This function needs two geometric predicates: left_turn(p, q, r) returns true if the point sequence p, q, r forms a left turn, while rotate_ccw_less(p, q, r, s) returns true if the angle from the oriented segment pq to the oriented segment rs is less than π, which is equivalent to left_turn($p, q, p + (s - r)$). As a convenience, is_corner(h) returns left_turn(source(prev(h)), source(h), target(h)).

We note that it is possible that two pseudo-triangles share more than the original edge h (but then it is easy to see that they cannot share more than two). In this case, the reader can check that the algorithm does not miss the flip due to such (unavoidable) degeneracies.

An optimization we could have tried for the flip is to see if the two adjacent regions are triangles, which gives the diagonal without any geometric tests. (Note that because of the minimality of pseudo-triangulations, the union of these two triangles must be a convex quadrilateral.) There is no guarantee, however, that even a single edge is adjacent to two triangles (consider $p_1 = (0, -1)$, $p_2 = (0, 1)$, $p_k = (k, 0)$ for $k > 2$). Nevertheless, if the point set has h edges on the boundary of its convex hull, there are at least $h - 2$ triangles in the pseudo-triangulation (with equality iff all the pseudo-triangles are at most quadrangular).

4.3 Enumeration algorithm

Using the FINDPSEUDOFLIP function, it is easy to implement the recursive variant of the enumeration algorithm. As noted, the only variable to store in the recursion stack is the minimal edge e at the current node.

Since the number of pseudo-triangulations of n points grows exponentially with n, we will not be able to run the algorithm for values of n larger than, say, 20. In fact, $n = 10$, with up to 234,160 pseudo-triangulations, is already a challenge and takes on the order of the second. This dictates a few implementation choices.

First, the priority queue can be a simple vector of edges, sorted by Θ values, although using a binary heap is not a penalty and improves the performance slightly. Second, finding the flip without performing it saves a constant time. Third, a non-recursive version of the algorithm eliminates the function call overhead, which contributes a (small) constant factor overall. Fourth, storing the old diagonal e on the stack when a diagonal is flipped avoids searching for this edge by reverse flipping the edge e' in order to restore the original pseudo-triangulation. The second and fourth optimizations combined save 36% in runtime.

5 Experimental results

We started this investigation to support or find a counter-example to the conjecture 1. The conjecture is not know to be true even for small values of n. Our goal is to run our enumeration algorithm on Aichholzer et al.'s comprehensive database of point sets with cardinality $n \leq 10$ [4]. Our result is that for the over 14 million point sets S in the database up to $n \leq 10$, we have $\#T(S) \leq \#PT(S)$. Moreover, we also have computed the maximum number of pseudo-triangulations (Table 1) which enriches Aichholzer's compendium. Finally, we have packaged our software into a pseudo-triangulation workbench with which we can interactively examine pseudo-triangulations, flip edges, and perform various algorithms. This was extremely useful in exploring other conjectures, including about bounded-degree pseudo-triangulations.

In order to assert confidence in our implementations, we have independently devised two implementations of the enumeration algorithm, and checked that they agree on every point set in the database. The case $n = 10$ took about a month to compute on a cluster of 26 Sun workstations and another eight Pentium processors at 1 Ghz, and about 200 days for the independent computation by another co-author. Luckily, both results agreed!

Interestingly, it was observed by Oswin Aichholzer that, for $n = 8$, the maximum $\#PT$ was achieved for the *same two* point sets as for the maximum $\#T$, and conjectured that the same would be true for $n = 10$. Indeed, this is now verified. But this is not true for $n < 8$ nor for $n = 9$. At this time, it seems far-fetched to conjecture that, for all even $n \geq 8$, $\#PT(S)$ attain its maximum exactly for the point sets S which also maximize $\#T(S)$. Nevertheless, we can state:

Experimentally proven fact 15 *For any point set with $n \leq 10$, we have $\#T(S) \leq \#PT(S)$, with equality iff the point set is in convex position. In that case, $\#PT(S)$ is minimal.*

Table 1 shows the minimum and maximum values of $\#PT(S)$ for every value of n, and indicates the running time of our algorithms (both implementations were comparable). The point set with minimum $\#PT$ is the point set in convex position for $n \leq 10$. This contrasts sharply with the situation on triangulations [2]. In a previous version of this paper, we conjectured that the number of pseudo-triangulations is minimized for point sets in convex position for any value of n. In fact, this has been proven recently [5].

For $n = 11$, the point set database has recently been assembled by Aicholzer et al. and consists of 2,334,512,907 point sets. It is thus infeasible to compute the exact lower and upper bounds using this algorithm. Nevertheless, we can still compute the number of pseudo-triangulations for particular configurations (this s a further test of correctness of our algorithm, when the result is know mathematically), and on random point sets.

We conjectured that the ratio ρ of (blue and red) leaves to red leaves was bounded by 2. Experiments showed that this is simply not true. In Table 2, we display some lower bounds on ρ, as well as the best known upper bound.

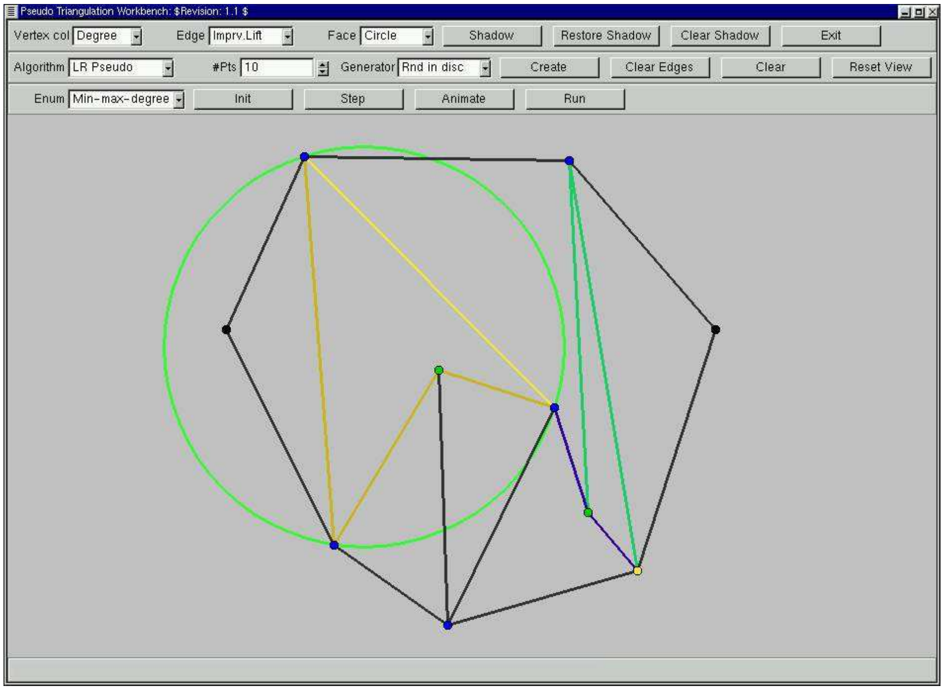

Figure 9: Screen shot of using the pseudo-triangulation workbench to test a conjecture

6 Conclusion

We have presented and implemented a new algorithm to enumerate all the pseudo-triangulations of a point set. This algorithm uses the theory of pseudo-triangulations that was developed for convex obstacles, in particular it makes reuse of the greedy flip algorithm.

Using the polytopal construction of [43], one could obtain another algorithm via the reverse-search paradigm [8]. Our algorithm is more general, however, since with the proper flip algorithm it also applies to matroids (in the dual, arrangements of pseudo-lines), while reverse search is limited to geometric systems.

The running time per triangulation is in theory $O(n^2)$, although it should be possible to lower that upper bound by using amortization of the flip algorithm, as well as better upper bounds on the ratio of leaves over red leaves. Also, the algorithm can be improved in theory using more fancy data structures, but since it is unlikely to be applied to point sets larger than 20, this is more of a theoretical exercise.

We independently developed two implementations of the algorithm, which agree on all point sets for $n \leq 10$. Using these implementation, we verified that Conjecture 1 is true for $n \leq 10$. The mathematical proof (even for such small values of n) is still waiting to be discovered.

Acknowledgement

We thank Ileana Streinu and the other participants of the NSF-supported Bellairs workshop on pseudo-triangulations for many enjoyable discussions.

# points	# point order types	lower-bound	upper bound		runtime
3	1	1	1	(1, #1)	$<$ 1sec
4	2	2	3	(1, #2)	$<$ 1sec
5	3	5	13	(2, #3)	$<$ 1sec
6	16	14	76	(8, #15)	$<$ 1sec
7	135	42	485	(30, #125)	1sec
8	3 315	132	3 555	(150, #2991 and #3199)	3min = 0.054 sec/order type
9	158 817	429	27 874	(774, #151 721)	990min = 0.374 sec/order type
10	14 309 547	1430	234 160	(4550, #13 413 894 and #13 812 360)	about 200 days

Table 1: Number of pseudo-triangulations found among all the order types. Between parentheses is $\#T(S)$ for the order type S maximizing $\#PT(S)$, followed by the index of S in the database.

# points	exact	lower-bound	upper bound
3	1		
4	2		
5	2.76923		4
6	3.49254		6
7	4.26786		8
8	4.89121		10
9	5.74258		12
10	6.28663		14
11	N/A	$\geq$ 6.11959	16
12	N/A	$\geq$ 5.709	18

Table 2: Maximum ratio ρ of (blue and red) leaves to red leaves during the enumeration. The second column is (a 5-digit approximation of) the exact number when available. The third column is a lower bound, obtained by trying random point sets with various distributions, while the fourth column is the best known upper bound.

References

[1] P. K. Agarwal, J. Basch, L. J. Guibas, J. Hershberger, and L. Zhang. Deformable free space tiling for kinetic collision detection. In B. Donald, K. Lynch, and D. Rus, editors, *New Directions in Algorithmic and Computational Robotics*, pp. 83–96. A. K. Peters, 2001.

[2] O. Aichholzer. Triangulation olympics. `http://www.cis.tugraz.at/igi/oaich/triangulations/counting/counting.html`

[3] O. Aichholzer. The path of a triangulation. In *Abstracts 13th European Workshop Comput. Geom.*, pp. 1–3, 1997.

[4] O. Aichholzer, F. Aurenhammer, and H. Krasser. Enumerating order types for small point sets with applications. *Order*, 19:265-281, 2002. Prelim. version in *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pp. 11–18, 2001.

[5] O. Aichholzer, F. Aurenhammer, H. Krasser, and B. Speckmann. Convexity minimizes pseudo-triangulations. *Comput. Geom.: Theory Appl.*, 28:3-10, 2004.

[6] A. M. Andrew. Another efficient algorithm for convex hulls in two dimensions. *Inform. Process. Lett.*, 9(5):216–219, 1979.

[7] P. Angelier and M. Pocchiola. A sum of squares theorem for visibility complexes. In *Aronov, Basu, Pach and Sharir (eds.), Discrete and Computational Geometry - The Goodman-Pollack Festschrift*, Algorithms and Combinatorics, Springer Verlag, Berlin, pp. 79–139, 2003.

[8] D. Avis and K. Fukuda. Reverse search for enumeration. *Discrete Appl. Math.*, 65:21–46, 1996.

[9] S. Bereg. Transforming Pseudo-Triangulations *Inform. Process. Lett.*, 90(3):141–145, 2004.

[10] S. Bespamyatnikh. An efficient algorithm for enumeration of triangulations. *Comp. Geom. Theory Appl.* 23(3):271–279, 2002.

[11] A. Björner, M. Las Vergnas, N. White, B. Sturmfels, and G. M. Ziegler. *Oriented Matroids*. Cambridge University Press, Cambridge, 1993.

[12] H. Brönnimann. Designing and implementing a general purpose halfedge data structure. In *Proc. 5th Int. Workshop Algorithm Engineering.* (WAE), volume 2141 of *Lecture Notes Comput. Sci.*, Springer Verlag, pp. 51–66, 2001.

[13] 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.

[14] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, Germany, 2nd edition, 2000.

[15] E. D. Demaine, J. S. B. Mitchell, and J. O'Rourke, eds. Problem 40: The Number of Pointed Pseudotriangulations. In *The Open Problems Project*, `http://maven.smith.edu/~orourke/TOPP/P40.html`.

[16] M. Denny and C. Sohler. Encoding a triangulation as a permutation of its point set. In *Proc. 9th Canad. Conf. Comput. Geom.*, pp. 39–43, 1997.

[17] A. Fabri, G.-J. Giezeman, L. Kettner, S. Schirra, and S. Schönherr. On the design of CGAL, a computational geometry algorithms library. *Softw. – Pract. Exp.*, 30(11):1167–1202, 2000.

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

[19] L. J. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Trans. Graph.*, 4(2):74–123, 1985.

[20] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. *Comput. Geom. Theory Appl.*, 4:63–98, 1994.

[21] F. Hurtado, M. Noy, and J. Urrutia. Flipping edges in triangulations. *Disc. Comput. Geom.*, 22(3):333–346, 1999.

[22] L. Kettner. Using generic programming for designing a data structure for polyhedral surfaces. *Comput. Geom. Theory Appl.*, 13:65–90, 1999.

[23] L. Kettner, D. Kirkpatrick, Andrea Mantler, J. Snoeyink, B. Speckmann, and F. Takeuchi. Tight degree bounds for pseudotriangulations of points. *Comput. Geom. Theory Appl.* 25(1&2):1–12, 2003.

[24] L. Kettner, D. Kirkpatrick, and B. Speckmann. Tight degree bounds for pseudo-triangulations of points. In *Proc. 13th Canad. Conf. Comput. Geom.*, 2001.

[25] D. Kirkpatrick and J. Snoeyink. Computing common tangents without a separating line. In *Proc. 4th Workshop Algorithms Data Struct.* (WADS), volume 955 of *Lecture Notes Comput. Sci.*, pp. 183–193, Springer-Verlag, 1995.

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

[27] D. Kirkpatrick and B. Speckmann. Separation sensitive kinetic separation for convex polygons. In *Proc. Japan Conf Disc. Comp. Geom.* (JCDCG 2000), volume 2098 of *Lecture Notes Comput. Sci.*, pp. i244–251, Springer Verlag, 2001.

[28] D. E. Knuth. *Axioms and Hulls*, volume 606 of *Lecture Notes Comput. Sci.* Springer-Verlag, 1992.

[29] C. L. Lawson. Transforming triangulations. *Discrete Math.*, 3:365–372, 1972.

[30] D. T. Lee and F. P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks*, 14:393–410, 1984.

[31] Z. Li and S. Nakano. Efficient generation of plane triangulations without repetition. In *Proc. 28th Int. Colloq. Automata, Languages and Programming* (ICALP), volume of *Lecture Notes Comput. Sci.*, pp. 433-443, 2001.

[32] Peter McMullen. Modern developments in regular polytopes. In T. Bisztriczky, P. McMullen, R. Schneider, and A. Ivić Weiss, editors, *Polytopes: Abstract, Convex and Computational*, volume 440 of *NATO ASI Series*, pages 97–124. Kluwer Academic Publishers, 1994.

[33] S. Negami. Diagonal flips of triangulations on surfaces. In *Yokohama Math. J.* 47:1–44, 1999.

[34] M. H. Overmars and J. van Leeuwen. Dynamically maintaining configurations in the plane. In *Proc. 12th Annu. ACM Sympos. Theory Comput.*, pp. 135–145, 1980.

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

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

[37] M. Pocchiola and G. Vegter. The visibility complex. *Internat. J. Comput. Geom. Appl.*, 6(3):279–308, 1996.

[38] M. Pocchiola and G. Vegter. On polygonal covers. In B. Chazelle, J. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, volume 223 of *Contemporary Mathematics*, pp. 257–268. AMS, Providence, 1999.

[39] M. Pocchiola. Horizon trees versus pseudo-triangulations. In *Abstracts 13th European Workshop Comput. Geom.*, page 12, 1997.

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

[41] D. Randall, G. Rote, F. Santos, and J. Snoeyink. Counting triangulations and pseudo-triangulations of wheels. In *Proc. 13th Canad. Conf. Comput. Geom.*, pp. 149–152, 2001.

[42] S. Ray and R. Seidel. A Simple and Less Slow Method for Counting Triangulations and for Related Problems. *Proc. European Workshop Comput. Geom.*, Seville, pp. , 2004.

[43] G. Rote, I. Streinu, and F. Santos. Expansive motions and the polytope of pointed pseudo-triangulations. In *Aronov, Basu, Pach and Sharir (eds.), Discrete and Computational Geometry - The Goodman-Pollack Festschrift*, Algorithms and Combinatorics, Springer Verlag, Berlin, pp. 699-736, 2003.

[44] R. Seidel and F. Santos. A better upper bound on the number of triangulations of a planar point set. *J. Combin. Theory Ser. A*, 102(1):186-193, 2003.

[45] D. D. Sleator, R. E. Tarjan, and W. P. Thurston. Rotation distance, triangulations, and hyperbolic geometry. *J. Amer. Math. Soc.*, 1:647–682, 1988.

[46] I. Streinu. A combinatorial approach to planar non-colliding robot arm motion planning. In *Proc. 41st Annu. IEEE Sympos. Found. Comput. Sci.* IEEE, pp. 443–453, 2000.

[47] W. T. Tutte. A census of planar triangulation. *Canad. J. Math.*, 14:21–38, 1962.