Computing Visibility Graphs via Pseudo-triangulations

Michel POCCHIOLA
Gert VEGTER

LIENS - 95 - 15

Computing Visibility Graphs via Pseudo-triangulations

Michel POCCHIOLA
Gert VEGTER*

LIENS - 95 - 15

April 1995

Laboratoire d'Informatique de l'Ecole Normale Supérieure
45 rue d'Ulm 75230 PARIS Cedex 05

Tel : (33)(1) 44 32 00 00
Adresse électronique : ...@dmi.ens.fr

*Dept. of Math. and Comp. Sciences
University of Groningen
The Netherlands

Computing Visibility Graphs via Pseudo–triangulations *

Michel Pocchiola[†] Gert Vegter[‡]

April 1995

Abstract

We show that the k free bitangents of a collection of n pairwise disjoint convex plane sets can be computed in time $O(k + n \log n)$ and $O(n)$ working space. The algorithm uses only one advanced data structure, namely a splittable queue. We introduce greedy pseudo–triangulations, whose combinatorial properties are crucial for our method.

1980 Mathematics Subject Classification: 68Q25,68U05,52A10,52C10,52B10,05C90.
Key Words and Phrases: Visibility graphs, visibility complexes, triangulations, pseudotriangles, pseudo-triangulations, topological sorting, sweep algorithms.

---

*A preliminary version of this work appears in the Proceedings of the 11th Annu. ACM Sympos. on Comput. Geom. held in Vancouver, June 1995.

†Département de Mathématiques et d'Informatique, Ecole normale supérieure, ura 1327 du Cnrs, 45 rue d'Ulm 75230 Paris Cedex 05, France (pocchiola@dmi.ens.fr)

‡Dept. of Math. and Comp. Sc, University of Groningen P.O.Box 800, 9700 AV Groningen, The Netherlands (gert@cs.rug.nl)

Contents

1 Introduction

Visibility graphs (for polygonal obstacles) were introduced by Lozano-Perez and Wesley [16] for planning collision-free paths among polyhedral obstacles; in the plane a shortest euclidean path between two points runs via edges of the tangent visibility graph of the collection of obstacles, augmented with the source and target points. Since then numerous papers have been devoted to the problem of their efficient construction ([2, 6, 8, 9, 10, 12, 21, 24, 26, 27]) as well as their characterization (see [1, 5, 20] and the references cited therein).

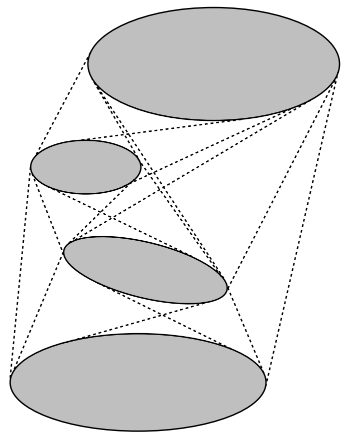


Figure 1: The visibility graph of a collection of four obstacles.

In this paper we present an optimal time and linear working space algorithm that computes the visibility graph of a collection $\mathcal{O}$ of n pairwise disjoint convex objects (= obstacles) in the plane. To the best of our knowledge, even for the case of line segments, this is the first optimal algorithm that uses linear working space. In [22, 23] we described an optimal time method for computing the so-called visibility complex of the collection $\mathcal{O}$ of obstacles. Just as the algorithm of Ghosh and Mount, see [8], it is based on a complicated data structure (namely the split-find structure of Gabow and Tarjan, see [7]). Therefore it is not suitable for a practical implementation. The algorithm presented in this paper is suitable for a practical implementation, for it uses a rather simple advanced data structure, namely the splittable queue. Recall that a *bitangent* is a closed line segment whose supporting line is tangent to two obstacles at its endpoints; it is called *free* if it lies in *free space* (i.e., the complement of the union of the relative interiors of the obstacles). An *exterior* (*interior*) bitangent is a bitangent lying on the boundary of (in the interior of) the convex hull of the collection $\mathcal{O}$ of obstacles. The endpoints of these bitangents subdivide the boundaries of the obstacles into a sequence of arcs; these arcs and the free bitangents are the edges of the visibility graph of the collection of obstacles, see Figure 1. Our main result is the following.

Theorem 1 *Let B be the set of free bitangents of a collection $\mathcal{O}$ of n pairwise disjoint obstacles, and let k be the cardinality of B. There is an algorithm that computes the set B in $O(k + n \log n)$ time and $O(n)$ working space—under the assumption that the bitangents between two obstacles are computable in constant time. Furthermore, if desired, the algorithm can compute the visibility graph (or the visibility complex) of the collection of obstacles in the same space and time bounds.*

Our approach is to turn B into a poset (partially ordered set) $(B, \prec)$ and to compute a linear extension of $(B, \prec)$, i.e., to embed $\prec$ into a linear (total) order. In other words, we solve the topological sorting problem for $(B, \prec)$, see [13, 14].

To define this partial order, we first introduce some terminology. The set of unit vectors in the plane is the 1-sphere $\mathcal{S}^1$. Let $exp : \mathcal{R} \to \mathcal{S}^1$ be the universal covering map of the 1-sphere, defined by $exp(u) = (\cos u, \sin u)$. Furthermore, let $B^{\rm or}$ be the oriented version (double cover) of B, obtained by associating with each $b \in B$ the two *directed* versions of b. For a directed bitangent $b \in B^{\rm or}$ let $v(b) \in \mathcal{S}^1$ be the unit vector along the directed line supporting b. We define X_0 to be the set $\bigcup_{b \in B^{\rm or}} \{b\} \times exp^{-1}(v(b))$. A point v in X_0 is called a *bitangent* in X_0; its first component is a bitangent in B, denoted by $\mathrm{bit}(v)$, and its second component is called its *slope*, denoted by $\mathrm{Slope}(v)$. We identify a bitangent in B with the corresponding bitangent in X_0 with slope in $[0, \pi[$. Two bitangents v and v' in X_0 are *crossing, disjoint,* etc., if the corresponding bitangents $\mathrm{bit}(v)$ and $\mathrm{bit}(v')$ in B are crossing, disjoint, etc.

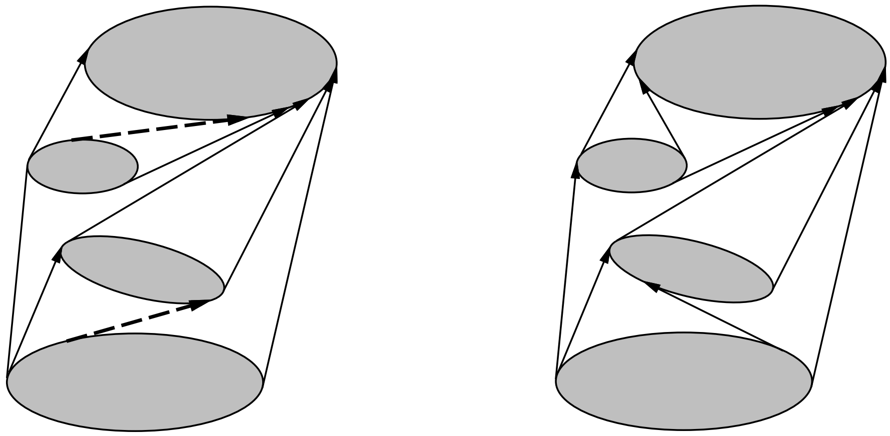
The (partial) order $\prec$ on X_0 is defined as follows: $b \prec b'$ if there is a counterclockwise oriented curve joining (some point of) $\mathrm{bit}(b)$ to $\mathrm{bit}(b')$, that runs along the edges (arcs and bitangents) of the visibility graph of the obstacles, and that sweeps an angle of $\mathrm{Slope}\ b' - \mathrm{Slope}\ b$ (a more formal definition is given in Section 2). This order has several nice properties, on which our algorithm is based. At this point we just mention that two crossing bitangents are comparable with respect to $\prec$ (see Lemma 3). Since $\prec$ is compatible with the slope-order on X_0, an obvious extension of $\prec$ is the linear order obtained by sorting the elements of X_0 according to increasing slope. However, this is computationally too expensive. To obtain the proper setting for dealing with the problem of extending $\prec$ to a linear order on X_0, we use the notion of filter [1]. A special type of filter of X_0 is the subset of bitangents $I(u)$, defined for $u \in \mathcal{R}$, that consists of all bitangents in X_0 whose slope is greater than u. For each filter I of $(X_0, \prec)$ we define a maximal subset $G(I) = \{b_1, \cdots, b_m\}$ of I as follows: (1) b_1 is minimal in I, and (2) for $1 \leq i < m$, the bitangent b_{i+1} is minimal in the set of bitangents in I, disjoint from $b_1, b_2, \ldots, b_i$. Since crossing bitangents are comparable it follows that $G(I)$ is well-defined (independent of the choice of the b_i), and that $\min_\prec I \subseteq G(I)$. We shall prove that for each filter I the set $G(I)$ contains $3n - 3$ bitangents, that subdivide free space into regions called *pseudotriangles*. This subdivision, denoted by $G(I)$, is called a greedy pseudo-triangulation. The regions owe their name to their special shape, that will be explained in more detail in Section 2. We refer to Figure 2 for an example of greedy pseudo-triangulations associated with filters of X_0.

Our algorithm maintains the greedy pseudo-triangulation $G(I)$ as I ranges over a maximal chain of filters of the interval $[I(0), I(\pi)]$, viz the set of filters I with $I(0) \supseteq I \supseteq I(\pi)$. The basic operation that updates the pseudo-triangulation is a *flip* of a free bitangent, minimal in the filter. The key result is the following.

Theorem 2 *Let I be a filter of $(X_0, \prec)$ and let $b \in \min_\prec I$. Then $G(I \setminus \{b\})$ is obtained from $G(I)$ by flipping b, i.e., by replacing b with the only minimal bitangent in $I \setminus \{b\}$ disjoint from the other bitangents in $G(I)$ (see Figure 2).* □

If the obstacles are points, our method—translated into dual space—is an alternative for the topological sweep algorithm for arrangements of lines, of Edelsbrunner and Guibas, see [6]. Our pseudo-triangulations replace their (upper and lower) horizon trees.

[1] A filter I of a poset $(P, \prec)$ is a subset of P such that if $x \in I$ and $x \prec y$ then $y \in I$. The set of filters, ordered by reverse inclusion, is a poset. Our main interest in the notion of filters is that, given two filters I and J with $J \subseteq I$ and $I \setminus J$ finite, the sequence $x_1, x_2, \ldots, x_k$ of elements of $I \setminus J$ is a linear extension of $(I \setminus J, \prec)$ if and only if the sequence of sets $I_1, I_2, \ldots, I_k$ defined by $I_i \setminus J = \{x_i, x_{i+1}, \ldots, x_k\}$, is an unrefinable chain of filters in the interval $[I, J]$. We borrow poset terminology from Stanley [25, Chap.3]. To keep the paper self-contained we review this terminology in Appendix C.



(a) (b)

Figure 2: (a) The greedy pseudo-triangulation associated with the filter $I(0)$ of bitangents with slope ≥ 0. The dashed bitangents b_1 and b_2 are both minimal in the filter $I(0)$. (b) The greedy pseudo-triangulation associated with the filter $I(0) \setminus \{b_1, b_2\}$ which is obtained from $G(I(0))$ by flipping b_1 and b_2.

The paper is organized as follows. In Section 2 we introduce the partial order $\prec$ on the set X of cells (or faces) of the visibility complex (whose X_0 is the set of vertices or 0-faces), we prove that this order satisfies a 'lattice-like' property, and we prove Theorem 2 by interpreting the greedy pseudo-triangulations as maximal antichains of $\prec$ on $X \setminus X_0$. In Section 3 we show how the flip operation can be efficiently implemented, using splittable queues.

2 The visibility complex

2.1 Terminology, pseudotriangle, pseudo–triangulations

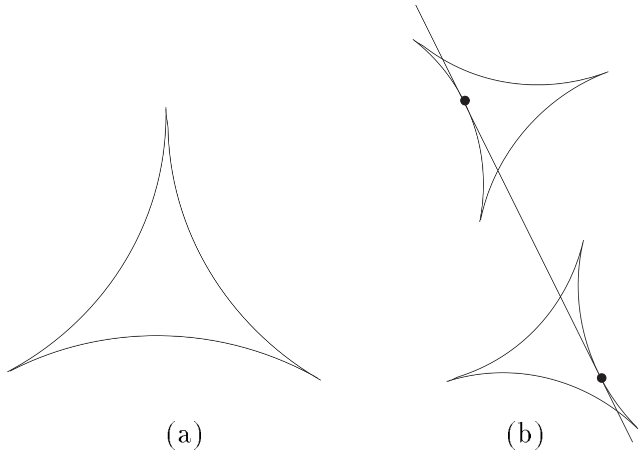


Figure 3: (a) A pseudotriangle. (b) Two disjoint pseudotriangles share exactly one common tangent line.

Let $O = O_1 \cup O_2 \cup \cdots \cup O_n$ be the union of n pairwise disjoint convex sets O_i (obstacles for short). We assume[2] that each obstacle is *strictly* convex, that is, the open line segment joining two of its points lies in its interior, and has a *smooth* boundary, that is, there is a well–defined tangent line through each of its boundary points. A *pseudo–triangulation* of a set of obstacles is the subdivision of the plane induced by a maximal (with respect to inclusion) family of pairwise noncrossing free bitangents. It is clear that a pseudo–triangulation always exists and that the bitangents of the boundary of the convex hull of the obstacles are edges of any pseudo–triangulation. A pseudo–triangulation of a collection of four obstacles is depicted in Figure 2. The subdivision owes its name to the special shape of its regions. A *pseudotriangle* is a simply connected subset T of the plane, such that (i) the boundary ∂T consists of three convex curves, that share a tangent at their common endpoint, and (ii) T is contained in the triangle formed by the three endpoints of these convex chains. These three endpoints will be called the *cusps* of T. At each boundary point of a pseudotriangle there is a well-defined tangent line, and there is a unique tangent line to the boundary of a pseudotriangle with a given unoriented direction (more formally the support function $\phi_T : \mathcal{S}^1 \to \mathcal{R}$ of T is well defined, continuous, and satisfies $\phi_T(u) = -\phi_T(-u)$).

Lemma 1 *The bounded free regions of any pseudo–triangulation are pseudotriangles. Furthermore the number of pseudotriangles (of a pseudo–triangulation of a collection of n obstacles) is $2n-2$, and the number of bitangents is $3n-3$.*

Proof. Let R be a family of noncrossing bitangents containing the bitangents of the boundary of the convex hull of the collection of obstacles. Assume that some free bounded face of the subdivision is not a pseudotriangle; from which 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 cusp points. In both cases we add to R a bitangent 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 cusp points

[2]This assumption is only for the ease of the exposition—in particular two bitangents in B are disjoint or intersect transversally (i.e., not at their endpoints).

of the exterior boundary but one. This curve contains a free bitangent not in R; hence R is not maximal.

An extremal point is a point on the boundary of an obstacle at which the tangent line to that obstacle is horizontal. Each pseudotriangle contains exactly 1 extremal point in its boundary (namely the touch point of the horizontal tangent line to the pseudotriangle). Since there are $2n-2$ extremal points in the interior of the convex hull of the obstacles there are exactly $2n-2$ pseudotriangles. The last result is then an easy application of the Euler relation for planar graphs. To see this observe that the set of vertices consists of all endpoints of bitangents. In particular every vertex has degree 3. Furthermore the number of edges, that lie on the boundary of some object, is equal to the number of vertices. Finally the total number of bounded regions is equal to the sum of the number of pseudotriangles and the number (n) of obstacles. □

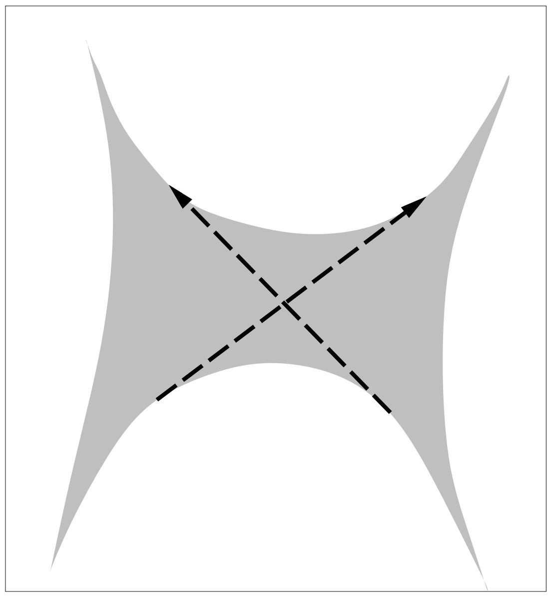


Figure 4: A pseudoquadrangle and its diagonals.

Lemma 2 *Let T and T' be two disjoint pseudotriangles. Then T and T' have exactly one common tangent line.*

Proof. For the existence part we apply the Intermediate Value Theorem to the continuous function defined as the difference between the support functions of T and T'. For the unicity we observe that tangent lines to a pseudotriangle cross inside the pseudotriangle. □

We will use this last lemma only in the case where T and T' are adjacent pseudotriangles (in a pseudo–triangulation). In that case the union of T and T' is called a *pseudoquadrangle*, and T and T' share two common bitangents called the diagonals of the pseudoquadrangle (see Figure 4).

2.2 Definition of the visibility complex as an abstract polytope

First it is convenient to identify the plane $\mathcal{R}^2$ with a 2-sphere $\mathcal{S}^2$ minus a point, called the point at infinity. Given a real number $u \in \mathcal{R}$ we set $C_u = 2\mathrm{CH}(O) + \mathcal{R}exp(u + \pi/2)$ (assuming without loss

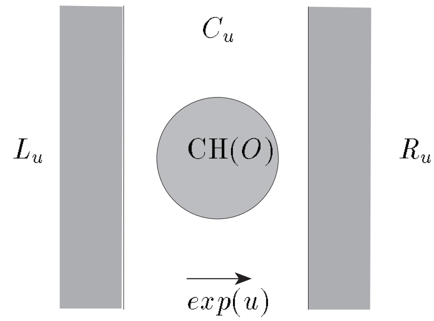


Figure 5: Free space $F_u = Cl(C_u \setminus O)$.

of generality that the origin is an interior point of the convex hull CH(O) of O), and we denote by L_u and R_u the two connected components of $\mathcal{R}^2 \setminus C_u$ (see Figure 5). The u-free space F_u is the closure of $C_u \setminus O$. A *ray* (p, u) is an element of $\mathcal{S}^2 \times \mathcal{R}$, consisting of a point p and a real number u. The point p is called the *origin* of the ray, and the real number u is called its *slope*. We denote by γ_{+i} (γ_{-i}) the set of rays (p, u) emanating from and tangent to obstacle O_i (i.e. $p \in \partial O_i$ and the tangent vector at p to O_i is $exp(u) \in \mathcal{S}^1$), that contain O_i in their left (right) half-plane; obviously $\gamma_{\pm i}$ is homeomorphic to $\mathcal{R}$. Let $C_i = O_i \times \mathcal{R}$, and let $C_- = \bigcup_{u \in \mathcal{R}} L_u \times \{u\}$ and $C_+ = \bigcup_{u \in \mathcal{R}} R_u \times \{u\}$. For a point p in $\mathcal{R}^2$ and a real number $u \in \mathcal{R}$ we are interested in the object (possibly L_u or R_u) that we can see from p in the direction $exp(u) \in \mathcal{S}^1$. This object is called the *view* from p along u, or the *forward view* from the ray (p, u) (the *backward view* from the ray (p, u) is the forward view of the *opposite* ray, $(p, -u)$). The view from a point p inside an object O_i is this object O_i, irrespective of the direction.

The underlying space $|X|$ of the visibility complex of O is the quotient space of the space of rays $\mathcal{S}^2 \times \mathcal{R}$ under the reflexive and transitive closure of the relation $\sim$, defined by: $(p, u) \sim (q, u)$ if and only if (1) the slope of the line (pq) is equal to u modulo π, and (2) the line segment $[p, q]$ lies in u-free space F_u. One can easily check that $|X|$ is homeomorphic to $\mathcal{S}^2 \times \mathcal{R}$. If we fix $u \in \mathcal{R}$ the set of rays in $|X|$ with slope u is a two-dimensional set, homeomorphic to $\mathcal{S}^2$. We shall refer to this set as the *cross-section* of $|X|$ at u. The *slope* of an equivalence class r, denoted by Slope(r), is the common slope of its rays, and we denote by seg(r) the set of origins of the rays in r. Observe that seg(r) is a maximal (with respect to the inclusion relation) free line segment, unless $r = \{(p, u)\}$ with p in the interior of some O_i (or L_u or R_u). By a slight abuse of terminology, an equivalence class will be called a *ray* in $|X|$. A ray r in $|X|$ is said to be tangent to obstacle O_i if the line segment seg(r) is tangent to O_i. It is convenient to denote by π the mapping which associates the ray $(p, u + \pi)$ with the ray (p, u). We stress that the rays $(p, u + k\pi)$, $k \in \mathcal{Z}$, are distinct points in $|X|$.

Observe that the canonical mapping from $\mathcal{S}^2 \times \mathcal{R}$ onto $|X|$, restricted to the interiors Inte(C_i) of C_i, with $i \in \{1, \ldots, n, +, -\}$, is one-to-one. The $n + 2$ canonical images of the sets Inte(C_i) and the $2n$ canonical images of the curves $\gamma_{\pm i}$ in $|X|$ induce a 3-dimensional cell (or face) decomposition of $|X|$. The 3–faces correspond to collections of rays with origins in the interior of the obstacles (including L_u and R_u), i.e., the Inte(C_i), with $i \in \{1, \ldots, n, +, -\}$. The 2–faces correspond to collections of rays with the same forward and backward views. The 1–faces correspond to collections of rays with the same forward and backward views and tangent to the same obstacle. The 0–faces correspond to collection of rays which are tangent to two obstacles.

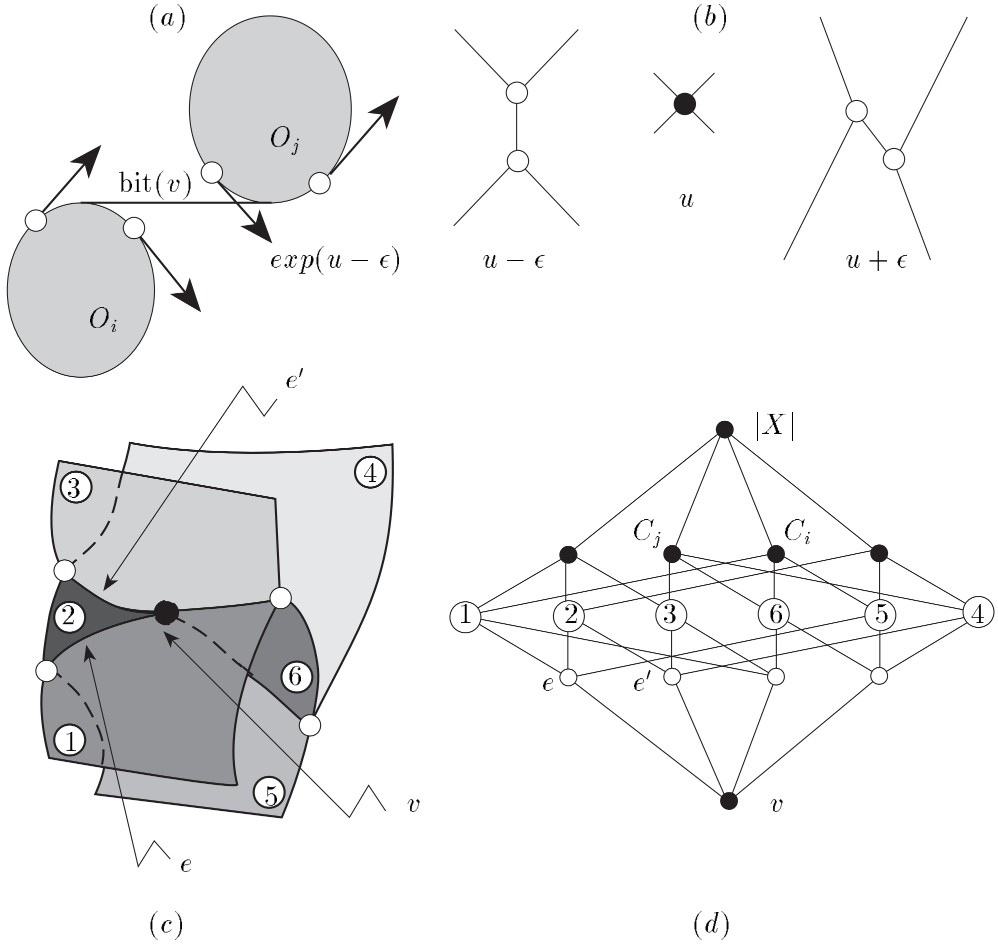


Figure 6: (a) Two obstacles defining a vertex v of the visibility complex with slope u. (b) (Local) cross-sections at slopes $u-\epsilon$, u and $u+\epsilon$. (c) Neighbourhood of a vertex of the visibility complex. (d) The Hasse diagram of the vertex-figure of a vertex of $P(X)$

We denote respectively by X_0, X_1, X_2, and X_3 the sets of 0-, 1-, 2-, and 3–faces of X. Let $P(X)$ be the poset of faces of X, augmented with $\emptyset$ and $|X|$, ordered by the inclusion relation of their closures. The local combinatorial structure of $P(X)$ is described in the following theorem, see Figure 6. (We refer to [3] for the terminology on abstract polytopes.)

Theorem 3 *$P(X)$ is an abstract polytope of rank 4. Furthermore the vertex-figure of a vertex is the face poset of a 3-dimensional simplex.* □

We denote by *arc* the onto mapping $x \mapsto \mathrm{arc}(x)$ from the set X_1 of edges of X onto the set of arcs of the visibility graph of O (more precisely $\mathrm{arc}(x)$ is the set of origins of the rays in x emanating from the object to which they are tangent). We denote by *bit* the onto mapping $x \mapsto \mathrm{bit}(x)$ from the set X_0 of vertices of X onto the set of free bitangents of O—The pre–image under the mapping bit of the bitangent $[p,q]$ with slope $u \in [0,\pi[$ is the set of rays $(p, u+k\pi)$, $k \in \mathcal{Z}$. An element of X_0 will occasionally be called a *bitangent* in X_0.

A *pseudo–triangulation* in X is a maximal (with respect to the inclusion relation) family of pairwise disjoint bitangents in X_0. Clearly, if G is a pseudotriangulation in X then (1) $\mathrm{bit}(G)$ is a pseudo–triangulation of the collection of obstacles, and (2) Card $G = 3n - 3$.

A face x is said to be *bounded* if $\mathrm{Slope}(x)$ is a bounded subset of $\mathcal{R}$, otherwise the face is said to be *unbounded*. The only unbounded faces are the 3–faces, and the 2–face that contains the rays whose origin is the point at infinity on $\mathcal{S}^2$.

Let x be a 1–face (viz an edge) or a bounded 2–face in X. We define $\sup x$ ($\inf x$) to be the ray with maximal (minimal) slope in the closure of x. The operator sup (resp. inf) is a one-to-one correspondance between the set of bounded 2—faces in X_2 and the set of vertices in X_0. We denote by inf (resp. sup) the inverse of the restriction of sup (inf) to the set of bounded faces. For a bounded 2–face x the vertices $\sup x$ and $\inf x$ subdivide the boundary of x into two curves, called the upper and lower boundary of the face. Observe also that the boundary of the unbounded 2–face has two connected components, denoted by γ_{+O} and γ_{-O}, that correspond to the set of rays emanating from and tangent to the boundary of the convex hull of the obstacles, that contain the convex hull of O in their left and right plane, respectively.

Remark 1 The mapping π is an automorphism of $P(X)$. One can easily check that $P(X)/\pi^2$ (which is finite) is still an abstract polytope (its 2-skeleton is called the *visibility complex* of O in [22, 23]). The rank–generating function (see [25]) of $P(X)/\pi^2$ is given by

$$F(q) = 1 + 2kq + 4kq^2 + (2k+1)q^3 + (n+2)q^4 + q^5. \tag{1}$$

where k is the number of free bitangents. This equality is a consequence of the previous discussion, viz on the bijection between the set of bounded 2–faces and the set of 1–faces, the shape of the vertex-figure, and the number $(n + 2)$ of 3–faces. The number of flags of $P(X)/\pi^2$ is 24 times the number of vertices, i.e., $48k$.

2.3 The poset $(X, \prec)$ and a local lattice–like property

Now we turn X into a poset $(X, \prec)$ by taking the transitive closure of the relation

$$\inf x \prec x \prec \sup x, \tag{2}$$

that is, for $t, t' \in X_0$ one has $t \prec t'$ if there exists a finite sequence of edges and/or 2–faces $x_1, \ldots, x_l$ in X such that (1) $t = \inf x_1$, (2) $\sup x_i = \inf x_{i+1}$, for $i = 1, \ldots, l-1$, and (3) $\sup x_l = t'$. Observe that we can replace each face that appears in the sequence $x_1, \cdots, x_l$ by the sequence of edges of its upper (or lower) boundary. In other words, $t \prec t'$ if there is a counterclockwise oriented curve in the plane from $\mathrm{bit}(t)$ to $\mathrm{bit}(t')$ that runs along the edges (arcs and bitangents) of the visibility graph of the obstacles (namely the arcs $\mathrm{arc}(x_i)$ and the bitangents $\mathrm{bit}(v_i)$ with $v_i = \inf x_i$, where we assume that x_i are edges), and which sweeps an angle of $\mathrm{Slope}(t') - \mathrm{Slope}(t)$. Clearly $\prec$ is compatible on X_0 with the slope order. It is convenient to adjoin a $\hat{0}$ and $\hat{1}$ to X with the convention that $\hat{0} \prec x \prec \hat{1}$ for all $x \in X$. We set $\sup\ x = \hat{1}$ and $\inf x = \hat{0}$ for all unbounded faces of X.

Observe that if two bitangents belong to the boundary of a pseudotriangle of some pseudo–triangulation then they are comparable. The same conclusion holds if the two bitangents are the diagonals of some pseudoquadrangle (viz the union of two adjacent pseudotriangles) of some pseudo–triangulation. From this observation we deduce a more general condition of comparability.

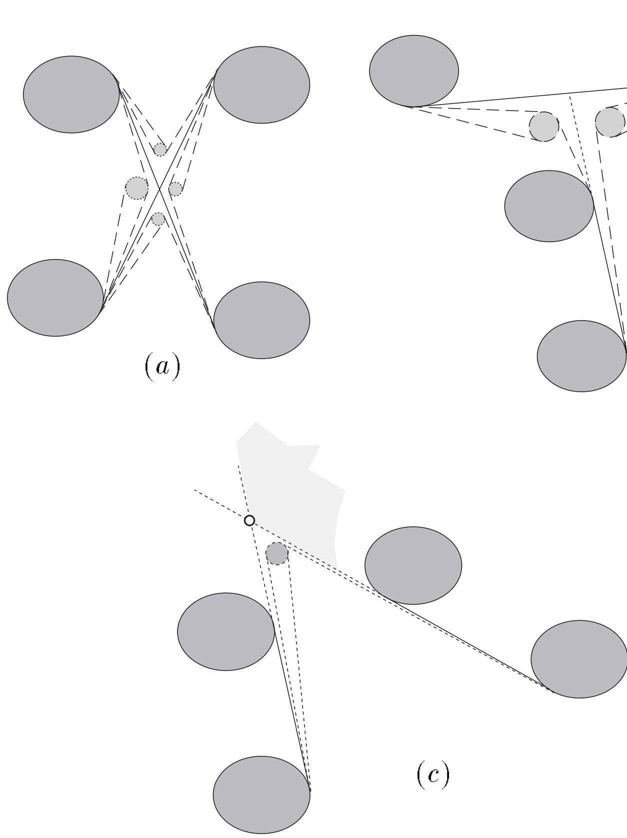


Figure 7: (a) bit(t) and bit(t') are crossing. The four added obstacles and the 12 added bitangents are shown dashed. (b) seg(t) $\setminus$ bit(t) and bit(t') are crossing. (c) seg(t) $\setminus$ bit(t) and seg(t') $\setminus$ bit(t') are crossing.

Lemma 3 *Let t and t' be two bitangents in X_0.*

1. *Assume that bit(t) and seg(t') are crossing. Then t and t' are comparable with respect to $\prec$.*

2. *Assume that seg(t) $\setminus$ bit(t) and seg(t') $\setminus$ bit(t') are crossing, say in point p, and that there is no free bitangent between p and the obstacles that lies in the wedge $t_+ \setminus t'_+$ (here t_+ is the open half-plane bounded by the supporting line of bit(t), that contains the line segment bit(t')). Then t and t' are comparable with respect to $\prec$.*

3. *$t \prec \pi^k(t')$, for all sufficiently large k.*

Proof. Assume first that bit(t) and bit(t') are crossing. Clearly it suffices to prove that bit(t) and bit(t') are the diagonals of a pseudoquadrangle of some pseudo–triangulation. To show the existence of a such pseudo–triangulation we add four sufficiently small obstacles near the crossing point of bit(t) and bit(t') as indicated in Figure 7a. Now we consider a pseudo–triangulation that contains the bitangent bit(t), and the $3 \times 4 = 12$ bitangents as indicated in the figure. Up to some flip operations we can assume that each of these four new obstacles contributes exactly 3 bitangents to the pseudo–triangulation. Removing these 4 added obstacles and their 12 bitangents yields a pseudo–triangulation (since the number of remaining bitangents is $3n-3$) with the desired property. A similar construction yields the result in the case where $\mathrm{bit}(t)$ and $\mathrm{bit}(t')$ are disjoint (see Figure 7b, c). Now we prove claim (3). We can assume that $\mathrm{bit}(t)$ and $\mathrm{bit}(t')$ are disjoint. Consider a pseudotriangulation G that contains $\mathrm{bit}(t)$ and $\mathrm{bit}(t')$, and consider a curve that joins $\mathrm{bit}(t)$ and $\mathrm{bit}(t')$. This curve crosses a finite sequence of bitangents in G, say $b_1, b_2, \ldots, b_l$. Let $t_j \in X_0$ such that $\mathrm{bit}(t_j) = b_j$, with $t_0 = t$ and $t_l = t'$. Since t_j and t_{j+1} are bitangents in the boundary of a pseudotriangle (or both on the convex hull), they are comparable. Therefore $t_j \prec \pi^{k_j}(t_{j+1})$ for k_j sufficiently large. It follows that $t \prec \pi^k(t')$ for k sufficiently large ($k = \sum_j k_j$). □

Now we come to the *lattice–like* property. We denote by ϕ the one–to–one mapping

$$t \in X_0 \mapsto \sup \sup t \in X_0, \tag{3}$$

that is, $\phi(t)$ is the ray with maximal slope in the (closure of the) face for which t is the ray with minimal slope. On can easily check that $\phi \circ \pi = \pi \circ \phi$.

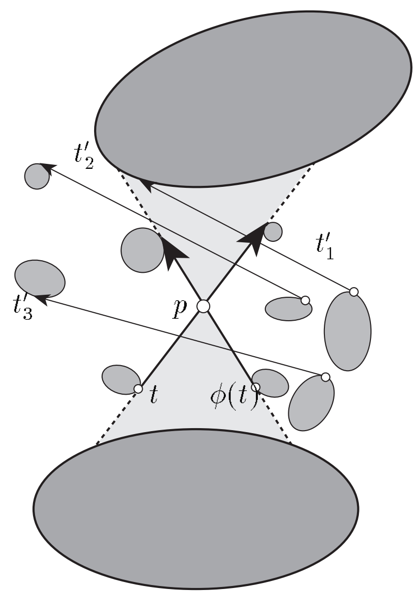


Figure 8: Illustration of the proof of the 'lattice-like' property.

Lemma 4 (Lattice-like property) *Let t and t' be two interior crossing bitangents in X_0 (i.e., bit(t) and bit(t') are crossing) with $t \prec t'$. Then $\phi(t) \preceq t'$ (and $t \preceq \phi^{-1}(t')$). In other words, $\phi(t)$ is the smallest bitangent in the set of bitangents crossing t and larger than t.*

Proof. Let p be the intersection point of $\mathrm{bit}(t)$ and $\mathrm{bit}(\phi(t))$, and let u and u^* be the slopes of t and $\phi(t)$, respectively. Let $t(\alpha) = (p, \alpha u + (1-\alpha)u^*)$, $\mathrm{seg}(t(\alpha)) = [a(\alpha), b(\alpha)]$

$$T = \bigcup_{\alpha \in [0,1]} \mathrm{seg}(t(\alpha)).$$

Clearly T is a subset of free space. Therefore the slope of t' is greater than the slope of $\phi(t)$, and $\mathrm{bit}(\phi(t))$ and $\mathrm{seg}(t')$ are crossing (first case), or $\mathrm{bit}(t')$ is tangent to the boundary of T (second case). See Figure 8 for an illustration. Hence it suffices to prove that t' and $\phi(t)$ are comparable with respect to $\prec$ in order to conclude that $\phi(t) \prec t'$. The first case is covered by Lemma 3, claim (1). In the second case $\mathrm{bit}(t')$ is tangent to the arc $\{b(\alpha) \mid \alpha \in]0,1[\}$, or to the arc $\{a(\alpha) \mid \alpha \in]0,1[\}$. Both cases are covered by claim (2) of Lemma 3. $\square$

Remark 2 Note that if t is an exterior bitangent then the set $\{\pi(t), \pi^2(t), \ldots\}$ is the set of bitangents greater than t and crossing t; its minimal element is $\pi(t)$.

2.4 Filters, antichains and greedy pseudo–triangulations

A *filter* is a *proper* (for our purpose) subset I of $X_0 \cup \{\hat{1}\}$ such that if $x \in I$ and $x \preceq y$, then $y \in I$. The set of all filters, ordered by reverse inclusion, form a (locally finite) poset. Our main interest in the notion of filter is due to its relation with the topological sorting problem: the sequence $\cdots b_{-2}b_{-1}b_0b_{+1}b_{+2}\cdots$ of bitangents in X_0 is a linear extension of $(X_0, \prec)$ if and only if the sequence $I_i = \{b_i, b_{i+1}, \ldots\}$ is a maximal chain of filters. For a finite subset A of X we define the filter A^+ by

$$A^+ = \{\, x \in X_0 \mid y \preceq x \text{ for some } y \in A \,\}. \tag{4}$$

The complement of A^+ in X_0 is denoted by A^-. For a filter I we let $\hat{I}$ be the subset of $X \setminus X_0$ defined by

$$\hat{I} = \{\, x \in X \setminus X_0 \mid \sup x \in I, \inf x \notin I \,\}. \tag{5}$$

Observe that the set of unbounded faces is a subset of $\hat{I}$.

Lemma 5 *The mapping $I \mapsto \hat{I}$ is a one-to-one correspondence between the set of filters of $(X_0, \prec)$ and the set of maximal antichains of $(X \setminus X_0, \prec)$, whose inverse is the map $A \mapsto A^+$.*

Proof. First we show that $\hat{I}$ is a maximal antichain of $(X \setminus X_0, \prec)$. Let $x \in \hat{I}$, $y \in X \setminus X_0$ with $x \prec y$, or $y \prec x$. Then $y \notin \hat{I}$. If $x \prec y$ we have $\sup x \preceq \inf y$ and, therefore, $\inf y \in I$, since $\sup x \in I$. This implies that $y \notin \hat{I}$. A similar conclusion holds if we assume that $y \prec x$. This proves that $\hat{I}$ is an antichain.

Now we prove that the antichain $\hat{I}$ is maximal. Let $x \in X \setminus X_0$ and consider the unrefinable chain $\{\, \sup^k x \mid k \in \mathcal{Z} \,\}$. This chain joins $X_0 \setminus I$ and I. Consequently this chain intersects $\hat{I}$, and x is comparable to an element in $\hat{I}$. Finally observe that $(\hat{I})^+ = I$, since (1) $\min I \subset (\hat{I})^+$ and (2) $(\min I)^+ = I$. $\square$

Theorem 4 *Let A be a maximal antichain in $(X \setminus X_0, \prec)$. Then*

1. *A depends only on its subset of 1–faces. More precisely, A is the union of the cofaces in $P(X)$ of its 1–faces. Furthermore, $P(A)$ is an abstract polytope of rank 3.*
2. *The numbers of 1–faces, 2–faces, and 3–faces in A are respectively $2n$, $3n$, and $n+2$ (and consequently $P(A)$ is spherical).*

Proof. Let x be an edge in A and let y be a 2–face incident to x. Clearly $\inf y \prec x \prec \sup y$, so $\inf y \in A^-$ and $\sup y \in A^+$. Therefore $y \in A$.

Conversely let y be a 2–face in A. Clearly its upper chain and lower chain are unrefinable, and join $\inf y \in A^-$ to $\sup y \in A^+$. Therefore these two chains intersect A. This proves claim (1).

The curves γ_i, $i \in \{\pm 1, \ldots, \pm n\}$, are edge-disjoint maximal chains, that together cover the set of edges of X. Therefore there is exactly one edge of the maximal antichain on each of these curves. Hence the number of edges in the antichain is $2n$. According to claim (1) the number of incidences between edges and 2–faces of a maximal antichain is 3 times the number of edges, and 2 times the number 2–faces. Therefore the number of 2–faces is $3n$. Planarity is proved by computing the Euler characteristic. $\square$

Let I be a filter and let $B_1(I), B_2(I), \ldots$ be the sequence of subsets of I defined by (1) $B_1(I)$ is the set of minimal bitangents in I, and (2) $B_{i+1}(I)$ is the set of minimal bitangents in the set of bitangents in I disjoint from the bitangents in $B_1(I), \ldots, B_i(I)$. Since the bitangents in $B_j(I)$ are pairwise non comparable they are pairwise disjoint, and consequently $\bigcup_{i\geq 1} B_i(I)$ is a pseudo–triangulation in X (in particular $B_i(I) = \emptyset$ for i sufficiently large). This pseudo–triangulation is denoted by $G(I)$ and is called the *greedy* pseudo–triangulation associated with the filter I [3]. Finally, for a filter I we define

$$S(I) = \{\, b \in I \mid \phi^{-1}(b) \notin I \,\}. \tag{6}$$

Now we come to the proof of Theorem 2, announced in the introduction. We give a slightly stronger form. For $Y \subset X_0$ we denote by Y_{int} (resp. Y_{ext}) the subset of Y consisting of interior (resp. exterior) bitangents.

Theorem 5 *1. For all filters I, and all interior (exterior) bitangents $b \in \min I$, the set difference $G(I \setminus \{b\}) \setminus G(I)$ is equal to $\{\phi(b)\}$ ($\{\pi(b)\}$).*

2. For all filters I, all bitangents $b \in G(I)$, and all $t \in I$ crossing b, one has $b \preceq t$.

3. For all filters I one has $G_{int}(I) = S_{int}(I)$.

Proof. First observe that claims (1) and (2) are obvious in the case where b is an exterior bitangent (see Remark 2). We show that (3) $\Rightarrow$ (1) $\Rightarrow$ (2), before proving claim (3).
(3) $\Rightarrow$ (1). First observe that $\phi(b)$ is disjoint from any $b' \in G(I) \setminus \{b\}$, otherwise $\phi(b)$ and b' are comparable, with $b' \prec \phi(b)$ (indeed $\phi(b) \prec b'$ implies that $\phi^{-1}(b') \in I$ and $b' \notin G(I)$). According to Lemma 4 this implies that $b' \preceq b$, a contradiction with $b \in \min I$. Therefore it is sufficient to prove that $\phi(b)$ is a bitangent in $G(I \setminus \{b\})$. Suppose the contrary holds. Then $\phi(b)$ intersects some $b' \in G(I \setminus \{b\})$, with $b' \prec \phi(b)$. But, according to Lemma 4, this implies that $b' \preceq b$, a contradiction.
(1) $\Rightarrow$ (2). Let $I_1, I_2, \ldots$ be the sequence of filters defined by $I_1 = I$ and $I_{k+1} = I_k \setminus B_1(I_k)$. Observe that if $b \in G(I_k) \setminus B_1(I_k)$ and $t \in I_k$ then $b \in G(I_{k+1})$ and $t \in I_{k+1}$. Therefore there exists a k such that $b \in B_1(I_k)$. From this we deduce that $b \preceq t$, since b is minimal in I_k.

Now we prove claim (3). We will prove successively that (i) $S_{int}(I) \subseteq G_{int}(I)$ (in particular the bitangents in $S_{int}(I)$ are pairwise disjoint), (ii) $G_{ext}(I) \subset S_{ext}(I)$ and Card $S_{ext}(I) =$ Card $G_{ext}(I) +$ 2, and (iii) Card $S(I) = 3n - 1$. These three properties imply that $G_{int}(I) = S_{int}(I)$, since Card $G(I) = 3n - 3$.

Let b be an interior bitangent. Then (*first case*) there is a $b' \in G(I)$ crossing b, with $b' \prec b$, or (*second case*) for all $b' \in G(I)$ crossing b one has $b \preceq b'$. In the first case Lemma 4 implies that $b' \preceq \phi^{-1}(b)$, and consequently that $b \notin S(I)$. In the second case b is smaller than any bitangent in $G(I)$ crossing it, therefore $b \in G(I)$. This proves claim (i).

[3]Observe that if $\prec_1$ is a total order on I compatible with $\prec$ on I then the set $G(I)$ can be enumerate as the sequence $b_1, b_2, \ldots, b_{3n-3}$ where (1) b_1 is the minimal bitangent in $(I, \prec_1)$,and (2) b_{i+1} is the minimal bitangent in $(I, \prec_1)$ disjoint from $b_1, b_2, \ldots, b_i$.

For an exterior bitangent t lying on γ_{+O} (γ_{-O}) we denote by $succ(t)$ the minimal exterior bitangent greater than t lying on γ_{+O} (γ_{-O}). Observe that $succ \circ \pi$ ($= \pi \circ succ$) is the restriction of ϕ to the set of exterior bitangents in X_0, and that the number h of exterior bitangents in B is defined by $succ^h = \pi^2$. Let t be the minimal element in I lying on γ_{+O}. Since $\phi^{-1}(t) = \pi^{-1} \circ succ^{-1}(t)$ it follows that t and $\pi(t)$ are both in $S(I)$. A similar result holds for the minimal element in I, say t', lying on γ_{-O}. Now we consider the sequence

$$t, succ(t), succ^2(t), \ldots$$

Clearly, if $succ^{j+1}(t) \in S(I)$ then $succ^j \in S(I)$. Therefore there is a k such that $succ^j(t) \in S(I)$ for $j = 0, 1, \ldots, k$ and $succ^j(t) \notin S(I)$ for $j > k$. Now observe that $\pi(t')$ lies on γ_{+O}. Therefore $succ^k(t) = \pi(t')$, since $\pi(t') \in S(I)$ and $succ(\pi(t')) = \phi(t') \notin S(I)$. Similarly, $succ^{k'}(t') = \pi(t)$, where k' is the greater index such that $succ^{k'}(t') \in S(I)$. It follows that $succ^{k+k'}(t) = \pi^2(t)$ and, consequently, that $k + k' = h$. Now observe that $G_{ext}(I)$ is a subset of

$$\{t, t', succ(t), succ(t'), succ^2(t), succ^2(t'), \ldots, \},$$

and that $succ^{j+1}(t) \notin G(I)$ if $succ^j(t) \notin G(I)$. Therefore, $G_{ext}(I) \subseteq S_{ext}(I)$, since $\pi(t)$ and $\pi(t')$ are not in $G_{ext}(I)$. Furthermore, a cardinality argument shows that $S_{ext}(I) = G_{ext}(I) \cup \{\pi(t), \pi(t')\}$. This proves claim (ii). Finally note that $S(I) = \sup \hat{I} = \sup \hat{I} \setminus X_1$, and consequently Card $S(I) = 3n - 1$, according to Theorem 4. This completes the proof. $\square$

3 The greedy flip algorithm and its analysis

3.1 The algorithm

For u in $\mathcal{R}$ we denote by $I(u)$ the filter of bitangents in X_0 with slope $\geq u$. Theorem 2 suggests a very simple algorithm : maintain the greedy pseudo-triangulation $G(I)$ while I describes a maximal chain of filters in the interval $[I(0), I(\pi)]$.

Greedy Flip Algorithm

1 compute the greedy pseudo-triangulation $G := G(I(0))$;
2 **repeat**
3 select a minimal bitangent b in G with slope less than π;
4 flip b; (i.e., replace b by $\phi(b)$ $(\pi(b))$ if b is an interior (exterior) bitangent.)

See Appendix D for an illustration of this algorithm on the configuration of Figure 1. Theorem 2 proves the correctness of this algorithm. Of course we still have to explain how to implement the flip operation (viz step 4) and how to select a minimal bitangent with slope less than π (viz step 3), so that the total cost of these operations is $O(k)$ time.

In Section 3.2 (and Appendix B) the construction of the initial pseudo–triangulation $G(I(0))$ is described in detail. Section 3.3 describes how to select a minimal bitangent. Section 3.4 describes an efficient implementation of the flip-operation, whose amortized cost is analyzed in Section 3.4.5.

3.2 Construction of the initial greedy pseudo–triangulation $G(I(0))$

Lemma 6 *The greedy pseudo–triangulation $G(I(0))$ of a collection of n disjoint convex obstacles in the plane can be computed in $O(n \log n)$ time.*

Proof. The construction is based on a standard rotational sweep à la Bentley–Ottmann, from direction 0 to direction π, during which we maintain the visibility map associated to the current direction. See Appendix B for more details. □

3.3 Minimal bitangents

Consider a filter I, a bitangent b in the greedy pseudo-triangulation $G(I)$, and a pseudotriangle T of $G(I)$. We denote by B_T the set of bitangents $t \in G(I)$ such that $\mathrm{bit}(t)$ appears in the boundary of T. The partial order $\prec$ restricted to B_T is a linear order. The minimal element of B_T is denoted b_T. We denote by $\mathrm{Ltri}(b)$ ($\mathrm{Rtri}(b)$) the pseudotriangle of $G(I)$ incident upon $\mathrm{bit}(b)$ and —locally—to the left (right) of $\mathrm{bit}(b)$, oriented along the direction of b. The *base-point* of T, denoted by p_T, is the tail of b_T, if $T = \mathrm{Rtri}(b_T)$, or the head of b_T, if $T = \mathrm{Ltri}(b_T)$. A subsegment of ∂T with counterclockwise (clockwise) orientation is called a *walk* (*reverse walk*) along ∂T. In particular, the walk starting at the base-point of T defines a linear order on the set of bitangents in $\mathrm{bit}(B_T)$, called the *slope order*, which coincides, via the mapping bit, with the linear order $\prec$ on B_T. We denote by b_+ (b_-) the minimal bitangent in $G(I)$ lying on γ_{+O} (γ_{-O}), if it exits.

Lemma 7 *Let I be a filter. Then an interior (exterior) bitangent b is minimal in I if and only if $b = b_{Rtri(b)} = b_{Ltri(b)}$ ($b = b_{Rtri(b)} = b_-$, or $b = b_{Ltri(b)} = b_+$).*

Proof. Indeed b is minimal in I if and only if the (two) edges $e \in X_1$ with $\sup e = b$ are in $\hat{I}$. $\square$

The successive cusps we pass during a walk starting at the base–point of T, are denoted by x_T, y_T and z_T. The *forward* and *backward* view of point p in ∂T are the points of intersection of ∂T with the tangent line at p, lying ahead and behind p, respectively. The point, whose forward (backward) view is p_T, if $T = \mathrm{Rtri}(b_T)$ ($T = \mathrm{Ltri}(b_T)$), is denoted by q_T, see also Figure 13.

For later use we isolate a simple, but crucial feature of pseudotriangles of greedy pseudo-triangulations.

Lemma 8 *Let T be a pseudotriangle of a greedy pseudo-triangulation.*
1. If $z_T \neq p_T$, then the part of ∂T between z_T and p_T is an arc.
2. If y_T lies between x_T and q_T, then the part of ∂T between y_T and q_T is an arc (i.e. it contains no bitangents).

Proof. We shall prove that no bitangent $t \in B_T$ has forward an backward views of smaller slope. This will prove 1, since all points on the segments $z_T p_T$ have both forward and backward view of smaller slope. A similar argument proves 2.

To prove the claim, suppose that both the backward and forward view, p_0 and p_1 say, of t have smaller slopes than t. We only consider the case in which p_0 has smaller slope than p_1, see Figure 9. Then $T = \mathrm{Ltri}(t)$, and the part of ∂T between p_0 and p_1 lies completely to the left of the line supporting t. Let t' be the bitangent between $T = \mathrm{Ltri}(t)$ and $\mathrm{Rtri}(t)$. The bitangent t' intersects

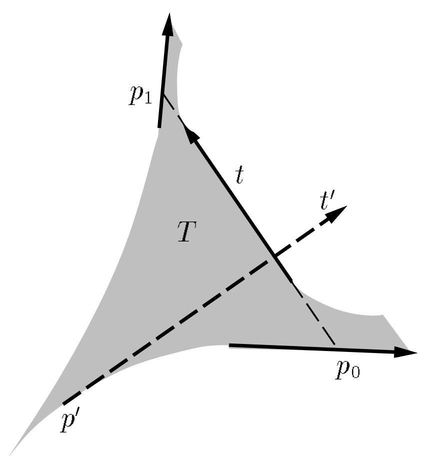


Figure 9: Forward and backward views p_0 and p_1 of t can't both have smaller slope than t.

t and its tail p' is a point on ∂T between p_0 and p_1, therefore its slope is less than the slope of t. But t and t' are crossing; and consequently $t' \prec t$ in contradiction with the greedyness of the pseudo-triangulation (claim 2 of Theorem 5). This proves the lemma. $\square$

3.4 Flipping minimal bitangents

3.4.1 The new pseudotriangles R' and L'

Consider a minimal bitangent b (with respect to some filter I), with $R = \mathrm{Rtri}(b)$ and $L = \mathrm{Ltri}(b)$. Let $b^* = \phi(b)$ be the bitangent obtained by flipping b. The right and left pseudotriangles of b^* (with respect to the filter $I \setminus \{b\}$) are denoted by R' and L', respectively. We denote by G and G' the pseudotriangulation $G(I)$ and $G(I \setminus \{b\})$, respectively. We consider the bitangent b_T for $T = R', L'$.

First consider the pseudotriangle R'. Let b'_R be the successor of b in B_R. The minimal element of $B_{R'}$ is one of the bitangents b'_R and b^*, viz the one with minimal slope. So $b^* = \min B_{R'}$, if $p^* = \mathrm{Tail}(b^*)$ lies between b and b'_R, and $b'_R = \min B_{R'}$, otherwise. Hence there are three basic cases, that will return throughout this section, see Figure 10.

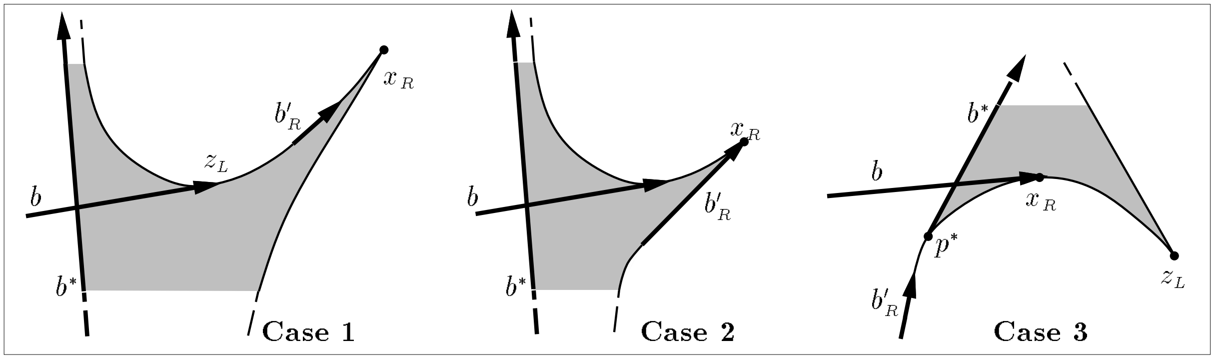


Figure 10: The pseudotriangle $R' = \mathrm{Rtri}b^*$ (shaded) is obtained by flipping bitangent b. Furthermore R' is the left or right triangle of b'_R (case 1 and 2, respectively), or $\partial R'$ doesn't contain b'_R (case 3).

Case 1 b and b'_R are not separated by a cusp of R.
Then $R' = \mathrm{Rtri}(b'_R)$, and p^* doesn't lie on the arc between b and b'_R. Therefore $\min B_{R'} = b'_R$.

Case 2 b and b'_R are separated by a cusp of R, but p^* doesn't lie on the arc between b and b'_R.
Then $R' = \mathrm{Ltri}(b'_R)$ and $\min B_{R'} = b'_R$. (Note: in this case $x_R = \mathrm{Head}(b'_R)$, as in Figure 10, or $x_R = \mathrm{Head}(b)$.)

Case 3 b and b'_R are separated by a cusp of R, but p^* lies on the arc between b and b'_R.
Then $R' = \mathrm{Rtri}(b^*)$ and $\min B_{R'} = b^*$.

The bitangent $\min B_{L'}$ is defined similarly.

We now consider the pseudotriangle R' in more detail, in particular its cusps $x_{R'}$, $y_{R'}$ and $z_{R'}$. (The story for L' is completely similar.)

Case 1 $R' = \mathrm{Rtri}(b'_R)$.
In this situation b and b'_R are not separated by a cusp, so $x_{R'} = x_R$. Furthermore, if p^* lies between x_R and y_R, then the second cusp $y_{R'}$ is equal to p^*, otherwise it is equal to y_R, see Figure 11a. Similarly the third cusp $z_{R'}$ is equal to y_L, if q^* lies between x_L and y_L, otherwise it is equal to q^*, see Figure 11b.

Case 2 $R' = \mathrm{Ltri}(b'_R)$ and $b'_R = \min B_{R'}$.
In this case the basepoint of R' is $\mathrm{Head}(b'_R)$, which lies between x_R and y_R. Therefore the first cusp $x_{R'}$ is equal to p^*, if p^* lies between x_R and y_R, otherwise it is equal to y_R, see Figure 11a. Similarly the second cusp $y_{R'}$ is equal to y_L, if q^* lies between x_L and y_L, otherwise it is equal to q^*, see Figure 11b. Finally the third cusp $z_{R'}$ is equal to z_L, if $\mathrm{Head}(b) = x_R$, otherwise it is equal to x_R, see Figure 11c.

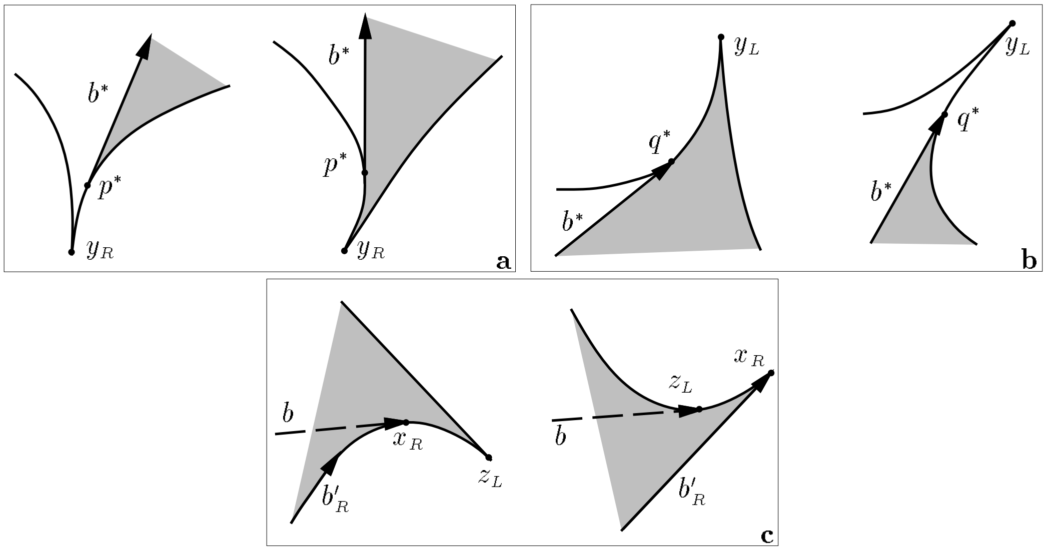


Figure 11: The cusps of R'.

Case 3 $R' = \mathrm{Rtri}(b^*)$ and $b^* = \min B_{R'}$.

In this case $\mathrm{Head}(b) = x_R$, and the tail p^* of b^* lies on the arc of ∂R separating b and b'_R. Therefore the basepoint of R' is p^*, which is also equal to the third cusp $z_{R'}$, see the left part of Figure 11a. Since in this case x_R is a cusp of R, the second cusp $y_{R'}$ is equal to z_L, see the left part of Figure 11c. Finally the first cusp $x_{R'}$ is equal to y_L or q^*, depending on whether q^* lies between y_L and z_L, or between x_L and y_L, see Figure 11b.

The table in Figure 12 summarizes the previous discussion.

	$x_{R'}$	$y_{R'}$	$z_{R'}$
Case 1	x_R	y_R or p^*	y_L or q^*
Case 2	y_R or p^*	y_L or q^*	z_L or x_R
Case 3	y_L or q^*	z_L	p^*

Figure 12: The cusps of R'.

3.4.2 The splittable queue `Awake[T]`

Conceptually the flipping can be done by walking—in positive direction, starting at the base–point—along the boundaries of the triangles L (left) and R (right) incident upon the flipped bitangent b, with one leg in every triangle, such that at any moment the tangent lines at the points underneath our left and right legs are parallel. We keep walking until these tangent lines coincide. At that point we have found b^*. This is too expensive, since some bitangents may be passed during many walks involved in the flip operations. To cut the budget, we shall need an auxiliary data structure, that enables us to start the walk at a more favorable point.

Observe that the tail p^* of b^* lies between the first cusp x_R and the point q_R, whose tangent contains the base–point Tail(b) of R. Similarly q^* lies between x_L and q_L. For a pseudotriangle T, a point in ∂T is called *awake* if it lies between x_T and q_T. Note that the points of ∂R that are awake have forward view of smaller slope, whereas the points awake in L have backward view of smaller slope, see Figure 13. Lemma 8 tells us that the set of points that are awake is a sequence of arcs and bitangents on a convex chain, possibly followed by a single arc between y_T and q_T (in case q_T does not lie between x_T and y_T).

If b and its successor b'_R in B_R are not separated by the cusp x_R, corresponding to case 1 in section 3.4.1, the point p^* lies even between q'_R and q_R, where q'_R is the point whose tangent contains Tail(b'_R), see Figure 13.

So the walk along ∂R starts at q'_R in case 1, and in x_R, otherwise. Similarly the walk along ∂L starts in q'_L or in x_L, where q'_L is the point on $\partial L'$ whose tangent contains Head(b'_L). Now x_T

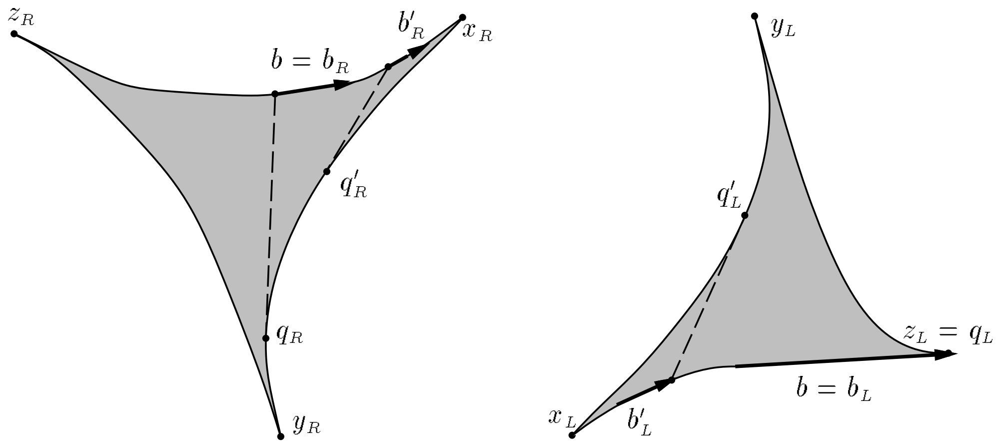


Figure 13: The set of points that are *awake* in T is the segment $x_T q_T$, for $T = L, R$. When the algorithm flips $b = b_R = b_L$, the walk on ∂T starts in q'_T (case 1), or in the cusp x_T (cases 2 and 3).

can be determined in $O(1)$ time, but how do we determine q'_T efficiently, for $T = L, R$? To this end we consider the segment $x_T q_T$ of points in ∂T, that are awake, as an alternating sequence of bitangents and arcs, or atoms for short, where the atoms are in slope order. This sequence will be represented by a *splittable queue*, denoted by `Awake[T]`, a data structure for ordered lists that allows for the following operations:

1. *enqueue* an atom, either at the head or at the tail of the list;
2. *dequeue* the head or the tail of the list;
3. *split* the sequence at an atom x; this split is preceeded by a *search* for the atom x.

A few comments on the split operation are in order. We assume that the initial search for the atom x is guided by a real-valued function, f say, defined for atoms in the sequence, that is monotonous with respect to the order of the atoms in the sequence. Now a *split* amounts to determining the atom x for which $f(x) = 0$, and successively splitting the sequence (destructively) into the subsequences of atoms with negative f–values and those with positive f–values. More specifically, to find the point q'_T (in case 1) we do a split operation in `Awake[T]`, where the search for q'_T is guided by the position of Tail(b'_T) with respect to the tangent lines at the endpoints of an atom. See section 3.4.3 for more details.

Lemma 9 *There is a data structure, implementing a splittable queue, such that an enqueue or dequeue operation takes $O(1)$ amortized time, and a split operation at an atom x on a queue of n atoms takes $O(\log min(d, n-d))$ amortized time, where d is the rank of x in the sequence represented by the queue.*

Moreover, a sequence of m enqueue, dequeue and split operations on a collection of n initially empty splittable queues is performed in $O(m)$ time.

For more details and a sketch of the proof we refer to appendix A.

We now describe in more detail (i) how to compute b^*, using `Awake[R]` and `Awake[L]`, and (ii) how to compute the queues `Awake[R']` and `Awake[L']`. Subsequently we prove that the total cost of (i) and (ii) amortizes to $O(k)$.

3.4.3 Construction of b^*.

If b and its successor b'_R in B_R are not separated by the cusp x_R of R (case 1), then during the construction of $b*$ the walk along ∂R starts in q'_R. In this case we *split* `Awake[R]` at q'_R into `AwakeMin[R]` and `AwakeMax[R]`, where the atoms in the former queue have smaller slope than the atoms in the latter queue. Otherwise, viz if b and b'_R are separated by the cusp x_R, we set `AwakeMin[R]` $\leftarrow \emptyset$ and `AwakeMax[R]` $\leftarrow$ `Awake[R]`. Here $\emptyset$ denotes the empty queue. In either case p^* lies on an arc, represented by an atom in the queue `AwakeMax[R]`. We similarly initialize the splittable queues `AwakeMin[L]` and `AwakeMax[L]`.

Now the simultaneous walk along ∂R and ∂L can be implemented by *dequeueing* atoms from `AwakeMax[R]` and `AwakeMax[L]`, until the atoms (arcs) are found that contain p^* and q^*, respectively. Obviously, this sequence of synchronous dequeue–operations takes time proportional to the number of dequeued atoms. So we construct b^* at the cost of at most one split on `Awake[R]` and at most one split on `Awake[L]`, followed by a number of successive dequeue operations.

We finally adjust the first atoms in the queues `AwakeMax[R]` and `Awake[L]`, (viz the atoms containing p^* and q^*, respectively), by replacing their endpoints of smaller slope with p^* and q^*, respectively. After this final operation the splittable queues `AwakeMax[R]` and `AwakeMax[L]` represent the segments p^*q_R of ∂R and q^*q_L of ∂L, respectively. We shall use these queues in the construction of the queues `Awake[R']` and `Awake[L']`.

3.4.4 Construction of `Awake[R']` and `Awake[L']`

To facilitate efficient maintenance of the collection of queues `Awake[T]`, for all pseudotriangles T, we also maintain the set of points of ∂T between the second cusp y_T and the third cusp z_T, that are not awake. These points are called *asleep*. They form a convex chain, viz the segment $y_T z_T$ or $q_T z_T$ of ∂T, depending on whether q_T lies between x_T and y_T or between y_T and z_T. This convex chain is also represented by a splittable queue `Asleep[T]`, whose atoms represent the arcs and bitangents of the chain in order of increasing slope.

We shall only describe the construction of the queues `Awake[R']` and `Asleep[R']` from the queues `AwakeMin[R]`, `AwakeMax[R]`, `Asleep[R]`, `AwakeMin[L]`, `AwakeMax[L]` and `Asleep[L]`, since the queues of L' are constructed similarly. In particular we show that this construction requires only a number of dequeue and *at most* 4 *enqueue* operations. Again we consider each of the cases, introduced in section 3.4.1, separately.

Case 1. $R' = \mathbf{Rtri}(b'_R)$**.**
Since in this case Head(b) is not a cusp of R, it is a cusp of L, see Figure 10, case 1. More precisely, Head(b) $= z_L$. Moreover, the point of ∂L whose tangent contains the basepoint Head(b) of ∂L, coincides with Head(b), so we also have $z_L = q_L$. In particular all points of ∂L between x_L and z_L are awake in L.

Furthermore, the basepoint of R' is Tail(b'_R), so we have $q_{R'} = q'_R$. Hence, by definition, all points that are awake in R' lie between x_R $(= x_{R'})$ and q'_R, so we set `Awake[R']` $\leftarrow$ `AwakeMin[R]`.

To see how `Asleep[R']` is constructed, first observe that b^* is asleep in R', since it lies on the segment $y_{R'}z_{R'}$ of $\partial R'$, beyond the point q'_R $(= q_{R'})$. We first describe how to initialize `Asleep[R']` so that it represents the chain of points on $q^*z_{R'}$, that are asleep in R'. To this end recall that `AwakeMax[L]` represents the segment q^*z_L of ∂L, since $q_L = z_L$, see the end of section 3.4.3. Furthermore, the segment y_Lz_L is a single arc, see Lemma 8. Therefore this arc is the last atom in `AwakeMax[L]`. So if $z_{R'} = y_L$, (cf the table in Figure 12), we initialize by setting `Asleep[R']` $\leftarrow$ `AwakeMax[L]`, after which we dequeue the last atom from this queue. If $z_{R'} = q^*$ the segment $q^*z_{R'}$ is empty, so we initialize by setting `Asleep[R']` $\leftarrow \emptyset$.

To complete the construction, observe that b^* is asleep in R'. Therefore we enqueue an atom representing b^* onto `Asleep[R']`, after which this queue represents the chain $p^*z_{R'}$. If $y_{R'} = p^*$ this completes the construction of `Asleep[R']`. So it remains to consider the case $y_{R'} = y_R$, see the table in Figure 12. The segment y_Rp^* is a single arc, see again lemma 8. If $q_{R'}$ $(= q'_R)$ lies between x_R and y_R, all points on the arc y_Rp^* are asleep in R', so the first atom of `Asleep[R']` should represent this arc. Finally, if $q'_R \in y_Rz_R$, the first atom of `Asleep[R']` should represent the arc q'_Rp^*. In either case we enqueue an atom at the head of `Awake[R']`, which represents an arc with end–point p^*, and begin–point y_R or q'_R. This completes the construction of `Asleep[R']` in this case.

Case 2. $R' = \mathbf{Ltri}(b'_R)$ **and** $b'_R = \min B_{R'}$**.**
Now b^* is *awake* in R'. The construction of the splittable queue `Awake[R']` from `AwakeMax[L]` is quite similar to the construction of `Asleep[R']` in case 2, although there is a slight difference in the construction of the tail of this queue. In this case R' is the *left* pseudotriangle of b'_R, so $q_{R'}$ is the tangent of $\partial R'$ that contains Head(b'_R). It is not hard to see that $q_{R'}$ lies on the segment q^*q_L of ∂L. But this segment is represented by `AwakeMax[L]`, see the end of section 3.4.3. So we start a *reverse* walk along ∂L, starting at q_L, until we have found $q_{R'}$. We know when to stop by considering the position of Head(b'_R) with respect to the tangent line in the current point of ∂L. This walk can be implemented by *dequeuing* atoms from the tail of `AwakeMax[L]` (cf the construction of b^*). When $q_{R'}$ is found, the queue `AwakeMax[L]` represents the segment $q^*q_{R'}$. So we set `Awake[R']` $\leftarrow$ `AwakeMax[L]`. The construction of `Awake[R']` is completed by enqueueing an atom representing b^*, followed by enqueing an atom representing the arc y_Rp^* in case $y_{R'} \neq p^*$.

It remains to describe the construction of `Asleep[R']`, viz the sequence of points of $\partial R'$ between $y_{R'}$ and $z_{R'}$ that are not awake. Note that all these points belong to ∂L.

If $z_{R'} = x_R$, we also know that $q_{R'} = z_R$. In this case none of the points of $\partial R'$ is asleep, so we set `Asleep[R']` $\leftarrow \emptyset$.

So suppose $z_{R'} = z_L$. In this case the base–point of L is x_R, see Figure 11c. We have already seen that $q_{R'}$ lies between q^* and q_L. So if $q_L \notin y_Lz_L$, then also $q_{R'}$ does not lie on y_Lz_L, and furthermore the latter segment coincides with the segment $y_{R'}z_{R'}$ of $\partial R'$. Therefore we set `Asleep[R']` $\leftarrow$ `Asleep[L]` in this case. If $q_L \in y_Lz_L$, then the segment y_Lq_L of ∂L is a single arc, see Lemma 8. So all atoms of `Asleep[R']`, except for the first one, are equal to the corresponding atoms of `Asleep[L]`. The first atom of `Asleep[R']` represents an arc that starts at either $y_{R'}$ or $q_{R'}$, depending on whether $q_{R'}$ lies on the arc between y_L and q_L or not (we can decide in $O(1)$ time which situation occurs). Hence we set `Asleep[R']` $\leftarrow$ `Asleep[L]`, after which we update the begin–point of the first atom as indicated.

Case 3. $R' = \mathbf{Ltri}(b'_R)$ **and** $b^* = \min B_{R'}$**.**
In this case R' is the *right* pseudotriangle of b^*, so p^* is the base–point of R', and $q_{R'} = p^*$, see Figure 10, case 3. Hence no point of $\partial R'$ is asleep, so we set `Asleep[R']` $\leftarrow \emptyset$. The construction of `Awake[R']` from `Asleep[L]` is similar to the construction of `Asleep[R']` from `Asleep[L]` in case 2. More precisely, if $q_L \notin y_L z_L$, we have $x_{R'} y_{R'} = y_L z_L$, which is represented by `Asleep[L]`. So set `Awake[R']` $\leftarrow$ `Asleep[L]` in this case.

If $q_L \in y_L z_L$, we also set `Awake[R']` $\leftarrow$ `Asleep[L]`, but we have to change the begin–point of the first arc from q_L into $y_{R'}$, which is either q^* or y_L, see the table in Figure 12.

3.4.5 Amortized complexity

As for the amortized time complexity, observe that the initial collection of splittable queues—one for each pseudotriangle in the greedy pseudotriangulation we start out with—can be computed in $O(n \log n)$ time (for instance simply by enqueueing the bitangents and arcs, that are awake in the boundary of each pseudotriangle). This amounts to $O(n)$ enqueue–operations. As we have just indicated, doing all flips and maintaining the collection of queues `Awake[T]` and `Asleep[T]`, $T \in \mathcal{T}$, cost $O(k)$ further enqueue, dequeue and split operations. Note that at any time the storage needed for all these queues is $O(n)$, see Lemma 1. Together with Lemma 9 this observation implies our main result, viz Theorem 1.

4 Conclusion

In this paper we have presented an optimal time and linear space algorithm for constructing the visibility graph of a set of pairwise disjoint convex obstacles in the plane. Our algorithm realizes a topological sweep of the visibility complex (a 2-dimensional cell complexe whose vertices correspond to the free bitangents of the collection of obstacles), and is based on new combinatorial properties of visibility graphs/complexes.

This work raises two questions that we intend to study in the future. The first question is whether our method can be extented to non-convex obstacles—It seems clear that the method can be extented to the computation of the visibility graph of the collection of relative convex hulls of non-convex obstacles (mainly because, in that case, free space remains decomposable into pseudotriangles); however the general case remains elusive. The second question is whether our algorithm can be turned into an algorithmic characterization of (some abstraction of) visibility graphs— as, for example, the greedy algorithm characterizes the independence set systems which are matroids (see [15]).

References

[1] P. K. Agarwal, N. Alon, B. Aronov, and S. Suri. Can visibility graphs be represented compactly? *Discrete Comput. Geom.*, 12:347–365, 1994.

[2] T. Asano, T. Asano, L. Guibas, J. Hershberger, and H. Imai. Visibility of disjoint polygons. *Algorithmica*, 1:49–63, 1986.

[3] T. Bisztriczky, P. McMullen, R. Schneider, and A. Ivić Weiss, editors. *Polytopes: Abstract, Convex and Computational*, volume 440 of *NATO ASI Series*. Kluwer Academic Publishers, 1994.

[4] B. Chazelle and H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *J. ACM*, 39:1–54, 1992.

[5] C. Coullard and A. Lubiw. Distance visibility graphs. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 289–296, 1991.

[6] H. Edelsbrunner and L. J. Guibas. Topologically sweeping an arrangement. *J. Comput. Syst. Sci.*, 38:165–194, 1989. Corrigendum in 42 (1991), 249–251.

[7] H. N. Gabow and R. E. Tarjan. A linear-time algorithm for a special case of disjoint set union. *J. Comp. Sys. Sci.*, 30:209–221, 1985.

[8] S. K. Ghosh and D. M. Mount. An output-sensitive algorithm for computing visibility graphs. *SIAM J. Comput.*, 20:888–910, 1991.

[9] L. J. Guibas and J. Hershberger. Computing the visibility graph of n line segments in $O(n^2)$ time. *Bull. EATCS*, 26:13–20, 1985.

[10] J. Hershberger. An optimal visibility graph algorithm for triangulated simple polygons. *Algorithmica*, 4:141–155, 1989.

[11] K. Hoffmann, K. Mehlhorn, P. Rosenstiehl, and R. E. Tarjan. Sorting Jordan sequences in linear time using level-linked search trees. *Inform. Control*, 68:170–184, 1986.

[12] S. Kapoor and S. N. Maheswari. Efficient algorithms for euclidean shortest paths and visibility problems with polygonal obstacles. In *Proc. 4th Annu. ACM Sympos. Comput. Geometry*, pages 178–182, 1988.

[13] D. Knuth. *The Art of Computer Programming: Fundamental Algorithms.* World Student Series Edition. Addison Wesley, second edition, 1973. 634 pages.

[14] Donald E. Knuth. ⟨⟨*Literate Programming*⟩⟩, volume 27 of *lectures Notes*. Center for Study of Language and Information, 1992.

[15] Bernhard Korte, László Lovász, and Rainer Schrader. *Greedoids*. Number 4 in Algorithms and Combinatorics. Springer-Verlag, 1991.

[16] T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM*, 22(10):560–570, 1979.

[17] 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.

[18] K. Mehlhorn. *Data Structures and Algorithms 1: Sorting and Searching*, volume 1 of *EATCS Monographs on Theoretical Computer Science*. Springer Verlag, 1984.

[19] B.M.E. Moret and H.D. Shapiro. *Algorithms from P to NP, volume I: Design and Efficiency.* Benjamin/Cummings, Redwood City, CA, 1990.

[20] J. O'Rourke. Computational geometry column 18. *Internat. J. Comput. Geom. Appl.*, 3(1):107–113, 1993.

[21] M. H. Overmars and E. Welzl. New methods for computing visibility graphs. In *Proc. 4th Annu. ACM Sympos. Comput. Geom.*, pages 164–171, 1988.

[22] M. Pocchiola and G. Vegter. The visibility complex. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 328–337, 1993.

[23] M. Pocchiola and G. Vegter. The visibility complex. *Internat. J. Comput. Geom. Appl.*, 1995. To appear.

[24] H. Rohnert. Time and space efficient algorithms for shortest paths between convex polygons. *Inform. Process. Lett.*, 27:175–179, 1988.

[25] Richard P. Stanley. *Enumerative Combinatorics*, volume 1. Wadsworth & Brooks/Cole, 1986.

[26] S. Sudarshan and C. P. Rangan. A fast algorithm for computing sparse visibility graphs. *Algorithmica*, 5:201–214, 1990.

[27] E. Welzl. Constructing the visibility graph of n line segments in the plane. *Inform. Process. Lett.*, 20:167–171, 1985.

A Splittable queues

Here we sketch a proof of Lemma 9. It is well known that finger trees suit our purpose, see e.g. [11], but even the much simpler red–black trees with father pointers will do. In this way we avoid the use of level links, which are rather complicated to maintain.

We augment a red–black tree with two special pointers, to the maximal and minimal atoms in the tree. Atoms (arcs) are stored in a leaf oriented fashion; they are represented by their endpoints and the tangent lines at their endpoints. In general, the same representation can be used to represent convex chains that are unions of atoms of the type just described. So we store a chain in a red-black tree in the following way: (i) store the atoms at the leaf; (ii) at an internal node, store the convex chain that is the union of all atoms in the subtree rooted at this leaf. This information is sufficient to guide the search for the atom at which we want to split the queue (chain), since the basic operation is to determine whether from a given point there is a tangent line to the convex chain. Furthermore the information at internal nodes can be maintained after rebalancing (recoloring, and a constant number of rotations).

The amortized $O(1)$ cost of the *enqueue* and *dequeue* operation follows from a standard argument, since it is well known that the amortized rebalancing cost of an insert operation on a red–black tree, i.e., the time spent *after* locating the father of the new node, is $O(1)$, see e.g. [18, chapter III], or [19, chapter 3.3]. Since upon enqueueing a new atom at the head or the tail of the list, the father of the new atom is either the maximal or the minimal node, it can be found in $O(1)$ time. One similarly shows that dequeueing takes $O(1)$ amortized time.

A similar argument holds for the split operation. Suppose we search for an atom x of rank d. By synchronously walking upward along the left and right ridge of the red–black tree, starting from the minimal and maximal node, we find the root of a subtree of height $O(\log \min(d, n-d))$ containing the atom x. Descending in this subtree, towards the leaf representing the atom x, takes $O(\log \min(d, n-d))$ time, after which we can do the actual split in $O(\log \min(d, n-d))$ time. The amortized time for rebalancing is also $O(\log \min(d, n-d))$.

To prove that a *sequence* of $O(m)$ operations on n initially empty splittable queues can be performed in $O(m)$ time, we provide each queue with $r - \log r$ credits, where r is the size (number of atoms) of the queue (we consider logarithms in base 2), see [4] for a similar analysis. Suppose that, due to a split operation, a queue of size r is split into two queues of size r_1 and r_2, where $r_1 \geq r_2$. To restore the credit invariant we deposit one additional credit for this split operation. Then the credits $r_1 - \log r_1$ and $r_2 - \log r_2$ for the new queues are available, since $2r_1 \geq r$ implies that:

$$r - \log r + 1 \geq (r_1 - \log r_1) + (r_2 - \log r_2) + \log r_2.$$

Restoring the credit invariants for the collection of queues upon an enqueue or dequeue operation is similar.

B Construction of a greedy pseudo–triangulation

This section is concerned with the proof of Lemma 6, that is, the construction in time $O(n \log n)$ of the greedy pseudo–triangulation $G = G(I(0))$. For simplicity assume that no free bitangent has slope 0. A useful aid in the construction of G is the *greedy visibility map* $M(u)$, associated with a slope $u \in [0, \pi]$. Let $B(u)$ be the bitangents in G with slope less than u. Note that $B(0) = \emptyset$, and $B(\pi)$ is the set of bitangents in G.

Every object O contains two points having a tangent line with slope u. These points are said to be of type *left* and *right* depending on whether the tangent line contains the object in its left or right half plane. The points are denoted by $O(u, \mathit{left})$ and $O(u, \mathit{right})$. The collection of all these points is denoted by $V(u)$.

Two distinct objects O and O' have exactly 8 common directed tangent lines. They form 4 pairs, denoted by (O, O', τ, τ'), where τ and τ' are either *left* or *right*. For instance, $(O, O', \mathit{left}, \mathit{right})$ is one of the two common tangent lines of O and O', containing O in its left half plane and O' in its right half plane.

From each point of $V(u)$ we shoot two rays, one with slope u, the other one with slope $-u$. We extend these rays until they hit an object, or a bitangent in the collection $B(u)$. In this way we partition free space into a number of regions that contain either one or two points of $V(u)$ in their boundary. These regions are called triangular and quadrangular, respectively. For convenience the two unbounded regions, in which we can walk in direction $u + \pi/2$ and $u - \pi/2$, respectively, are called quadrangular as well, even though they contain one point of $V(u)$ in their boundary.

If two triangular regions contain the same point p of $V(u)$ in their boundary, they are incident along one of the rays emerging from this point. We then merge these two regions by removing this ray. The point p is still the only point of $V(u)$ in the boundary of the merged region, which therefore is still triangular. The subdivision of free space that remains after removing all rays shared by triangular regions is called the *greedy visibility map* with respect to u. It is denoted by $M(u)$. Figure 14 depicts $M(u)$ for the initial direction $u = 0$, and the direction $u = \pi/2$. The greedy visibility map $M(0)$ coincides with what is usually called the horizontal visibility map of the collection $\mathcal{O}$. It can be constructed in $O(n \log n)$ time using a standard sweep line algorithm. Furthermore, the subdivision $M(\pi)$ is just the greedy pseudo–triangulation G (if we forget about the 4 unbounded faces that partition the complement of the convex hull). So we try to maintain $M(u)$ as u ranges over $[0, \pi]$

First we describe the construction of the sequence $B(\pi)$ of bitangents belonging to the pseudo–triangulation. Afterwards we briefly explain how to extend this method to maintain $M(u)$ as well. The appearance of a free bitangent corresponds to the disappearance of a quadrangular region. E.g. in the situation depicted in the lower left part of Figure 14 the topology of $M(u)$ will not change as u rotates beyond $\pi/2$, until u passes the slope of the bitangent contained in the quadrangular region labeled '6'. We represent the subdivision corresponding to the *quadrangular* regions of $M(u)$ by a directed graph $G(u)$, defined as follows.

Each quadrangular region of $M(u)$ contains two points of $V(u)$ in its boundary; We connect these points by drawing a path in this region that is increasing with respect to the direction $u + \pi/2$. In this way we obtain a directed plane graph $G(u)$, whose set of edges is in 1–1 correspondence with the set of quadrangular faces of $M(u)$, and whose vertices are the points of $V(u)$ in the boundary of the quadrangular faces, see Figure 14. There are two infinite edges, corresponding to the quadrangular faces that contain only one point of $V(u)$ in their boundary. The graph $G(0)$ contains $3n + 1$ edges, and $G(\pi)$ contains 4 edges. We shall see that there are $3n - 3$ events corresponding to the disappearance of an edge, and therefore to the appearance of a bitangent, cf

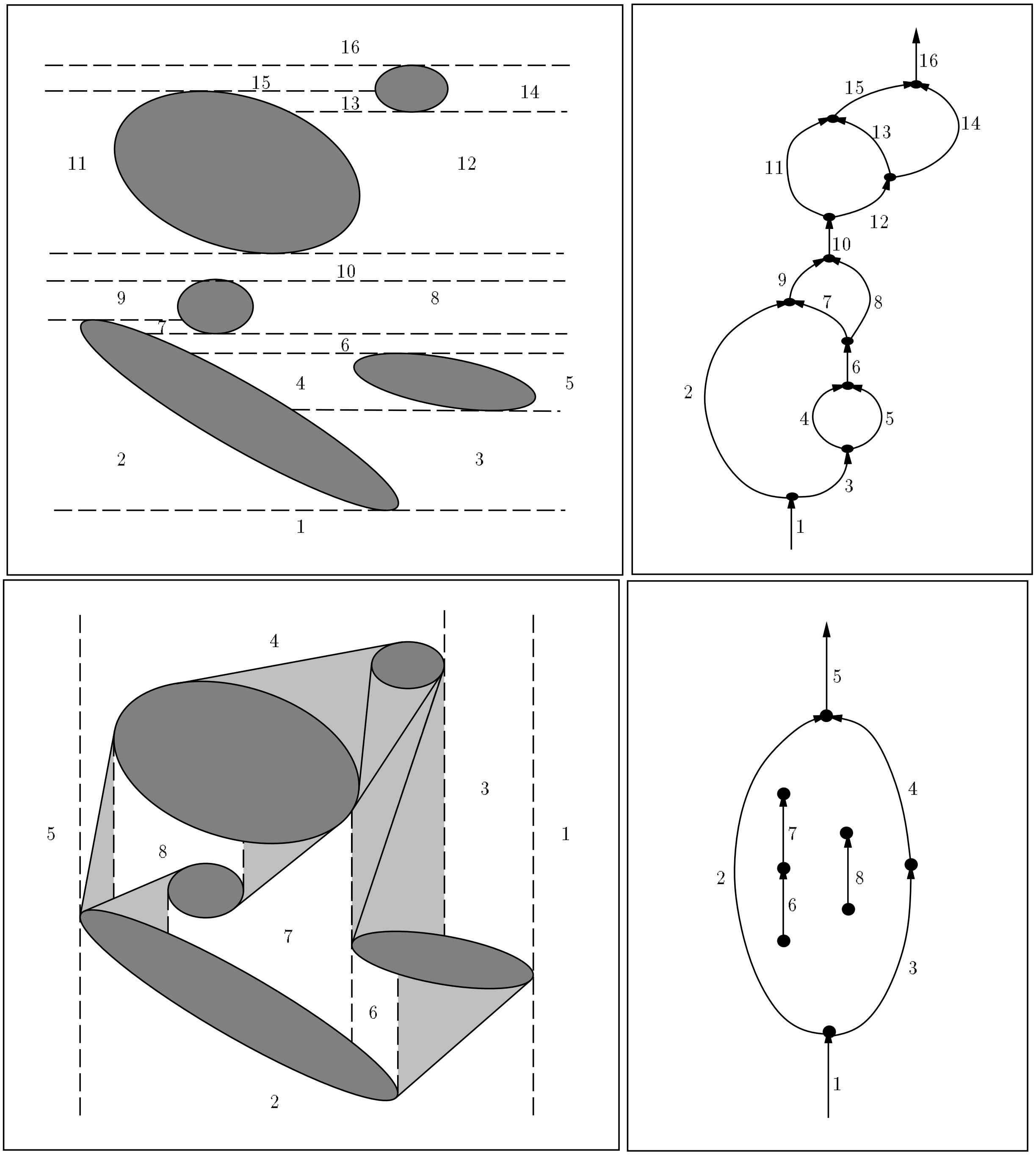


Figure 14: The visibility map: (a) Initial map (b) After ccw–rotation over $\pi/2$.

Lemma 1.

Consider now an edge e of the graph $G(u)$. Its endpoints are $O'(u, \tau')$ and $O''(u, \tau'')$. There are at most two tangent lines of type (O', O'', τ', τ''), whose slopes lies between 0 and u. Let $death(e)$ be the direction of these lines that is minimal, if this minimal element exists, or π otherwise, see Figure 15.

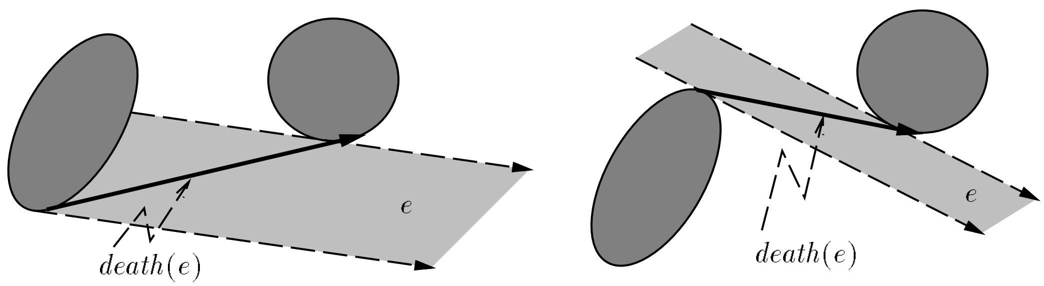


Figure 15: $death(e)$ is the critical direction associated with region e.

Let $\mathcal{D}(u)$ be the set of directions defined by

$$\mathcal{D}(u) = \{death(e) \mid e \text{ is an edge of } G(u) \text{ and } death(e) \text{ precedes } \pi\}.$$

The following obvious result is crucial for the correctness of our algorithm for the construction of the initial pseudo–triangulation.

Lemma 10 *Let the unit vectors u' and u'' be the directions of two consecutive elements of B.*

1. *The set $\mathcal{D}(u)$ does not change when u ranges over the open interval $(u'; u'')$.*

2. *The critical direction u'' is the minimal element of $\mathcal{D}(u)$, for u between u' and u''.*

We now describe the transition at the next critical direction, viz (i) updating the graph $G(u)$ when u passes this critical direction, and (ii) updating the set $\mathcal{D}(u)$. It is not hard to see that (i) takes $O(1)$ time, and (ii) takes $O(\log n)$ time, due to the maintenance of a priority queue. Figure 16 depicts a few cases.

We also describe the birth of pseudotriangles: the number of vertices of degree 3, plus the number of triangular regions, is invariant. This is obvious in the situations depicted in Figure 16. It also holds in the case where at least one of the regions a, b, c and d is triangular, see Figure 17. Note that the triangular regions *grow* during the sweep, so not all combinations of triangular and quadrangular regions are possible. For instance, in the topmost–leftmost part of Figure 16 it is not possible that region a is triangular whilst at the same time d is quadrangular, since in that case triangular region a doesn't grow: it shrinks near the edge along which it is incident with d. Finally the pseudo–triangulation $G(I(0))$ can easily be computed from the set of bitangents $B(\pi)$.

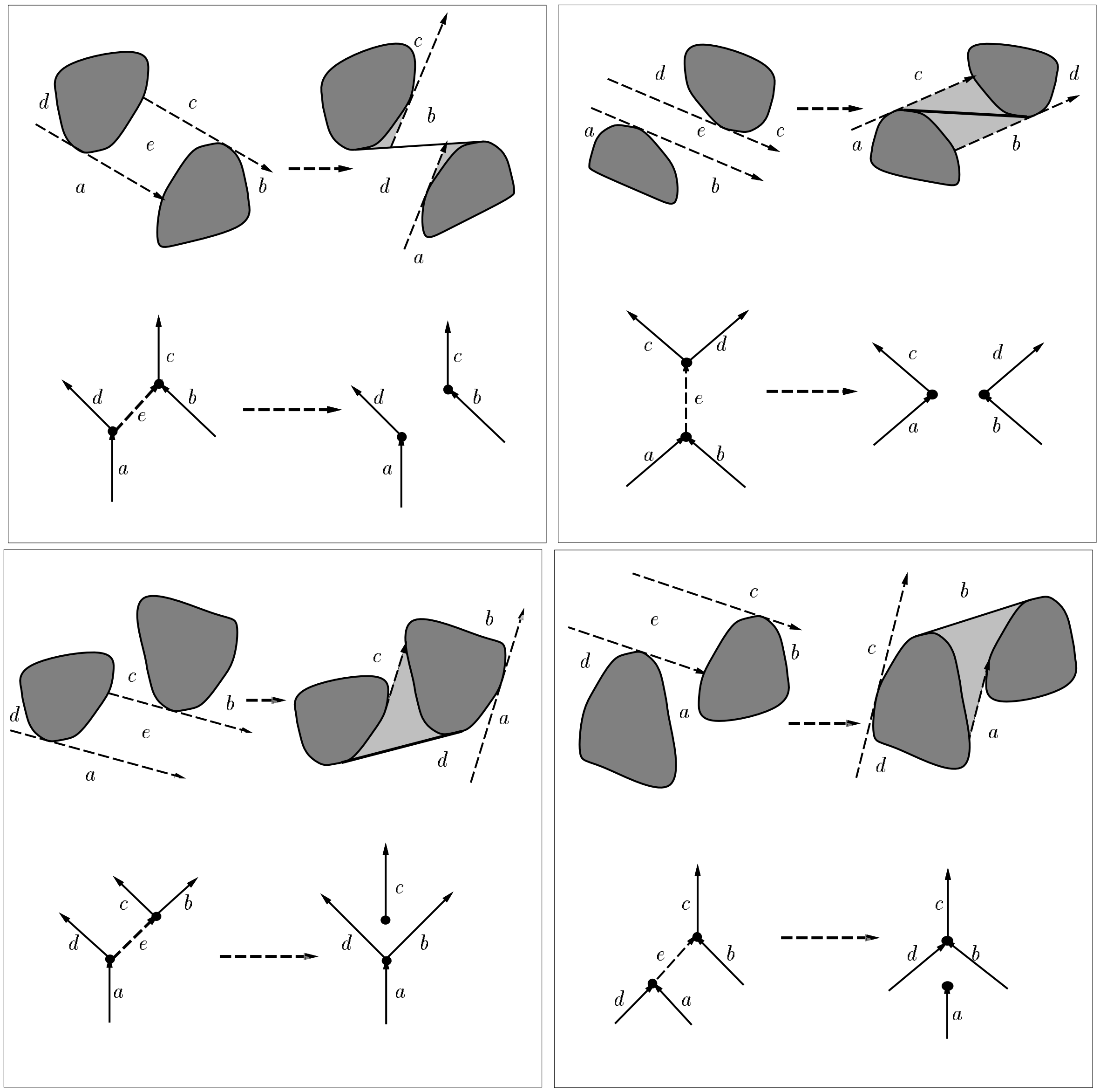


Figure 16: Transitions during the construction of the pseudo–triangulation.

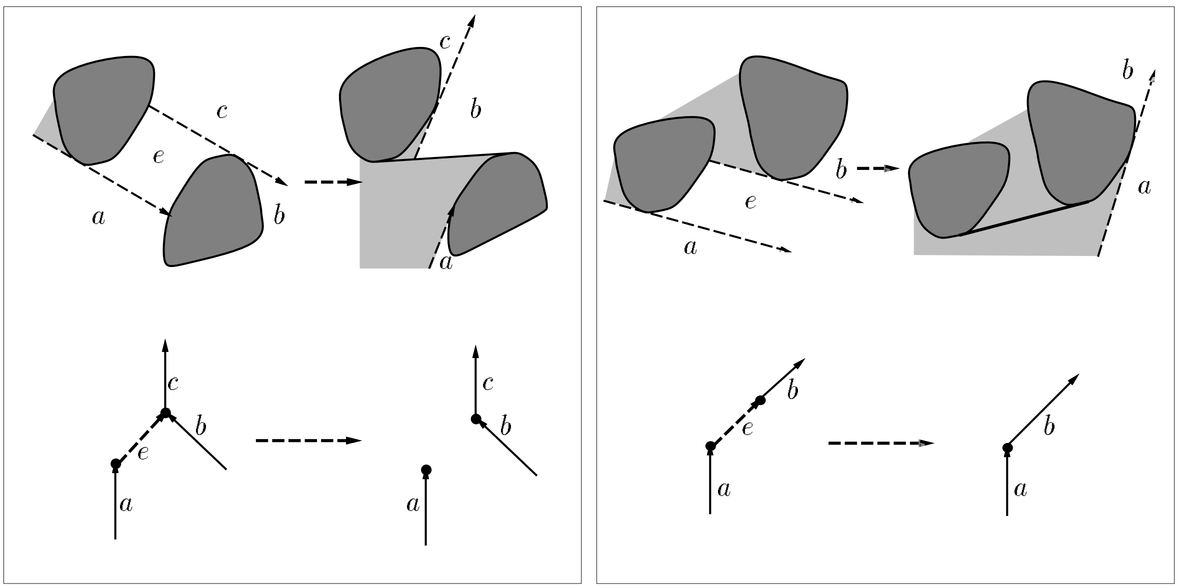


Figure 17: Transitions during the construction of the pseudo–triangulation: at least one of the regions a, b, c and d is triangular, and hence not represented by an edge in the graph $G(u)$.

C Poset terminology

In this section we review poset terminology, that we borrow from the book of Stanley [25, Chap. 3].

A *partially ordered set* P (or *poset*, for short) is a set, together with a partial order relation denoted by $\leq$. A *subposet* of P is a subset of P with the induced order. A special type of subposet is the *interval* $[x, y] = \{z \in P \mid x \leq z \leq y\}$. A poset P is called a *locally finite* poset if every interval of P is finite. If $x, y \in P$, then we say that y *covers* x if $x < y$ and if no element $z \in P$ satisfies $x < z < y$. The *Hasse diagram* of a poset is the graph whose vertices are the elements of P and whose edges are the cover relations.

A *chain* is a poset in which any two elements are comparable. A subset C of a poset P is called a *chain* if C is a chain when regarded as a subposet of P. The chain C of P is *saturated* (or *unrefinable*) if there does not exist $z \in P \setminus C$ such that $x < z < y$ for some $x, y \in C$ and such that $C \cup \{z\}$ is a chain. An *antichain* is a subset A of a poset P such that any two distinct elements of A are incomparable. A *filter* is a subset of P such that if $x \in I$ and $y \geq x$, then $y \in I$.

Finally we review the definition of an abstract polytope (see [17]).
An abstract n–polytope is a poset $(P, \leq)$, with elements called faces, which satisfies the following properties.

1. P has a unique minimal face F_{-1} and a unique maximal face F_n.
2. The flags (i.e. maximal chains) of P all contain exactly $n+2$ faces. Therefore P has a strictly monotone rank function with range $\{-1, 0, \ldots, n\}$. The elements of rank i are called the i-faces of P, or vertices, edges, and facets of P if $i = 0, 1$ or $n-1$, respectively.
3. P is strongly flag–connected, meaning that any two flags Φ and Ψ of P can be joined by a sequence of flags $\Phi = \Phi_0, \Phi_1, \ldots, \Phi_k = \Psi$, which are such that Φ_{i-1} and Φ_i are adjacent (differ by just one face), and such that $\Phi \cap \Psi \subset \Phi_i$ for each i.
4. Finally, if F and G are an $(i-1)$–face and an $(i+1)$–face with $F < G$, then there are exactly two i-faces H such that $F < H < G$. (Diamond Property.)

For a face F the interval $[F, F_n]$ is called the *coface* of P at F, or the *vertex–figure* at F, if F is a vertex.

D Example

In Figure 18 we depict the sequence of pseudo-triangulations, computed by the greedy-flip algorithm applied to the configuration of Figure 1, when at each step the bitangent with minimal slope is flipped.

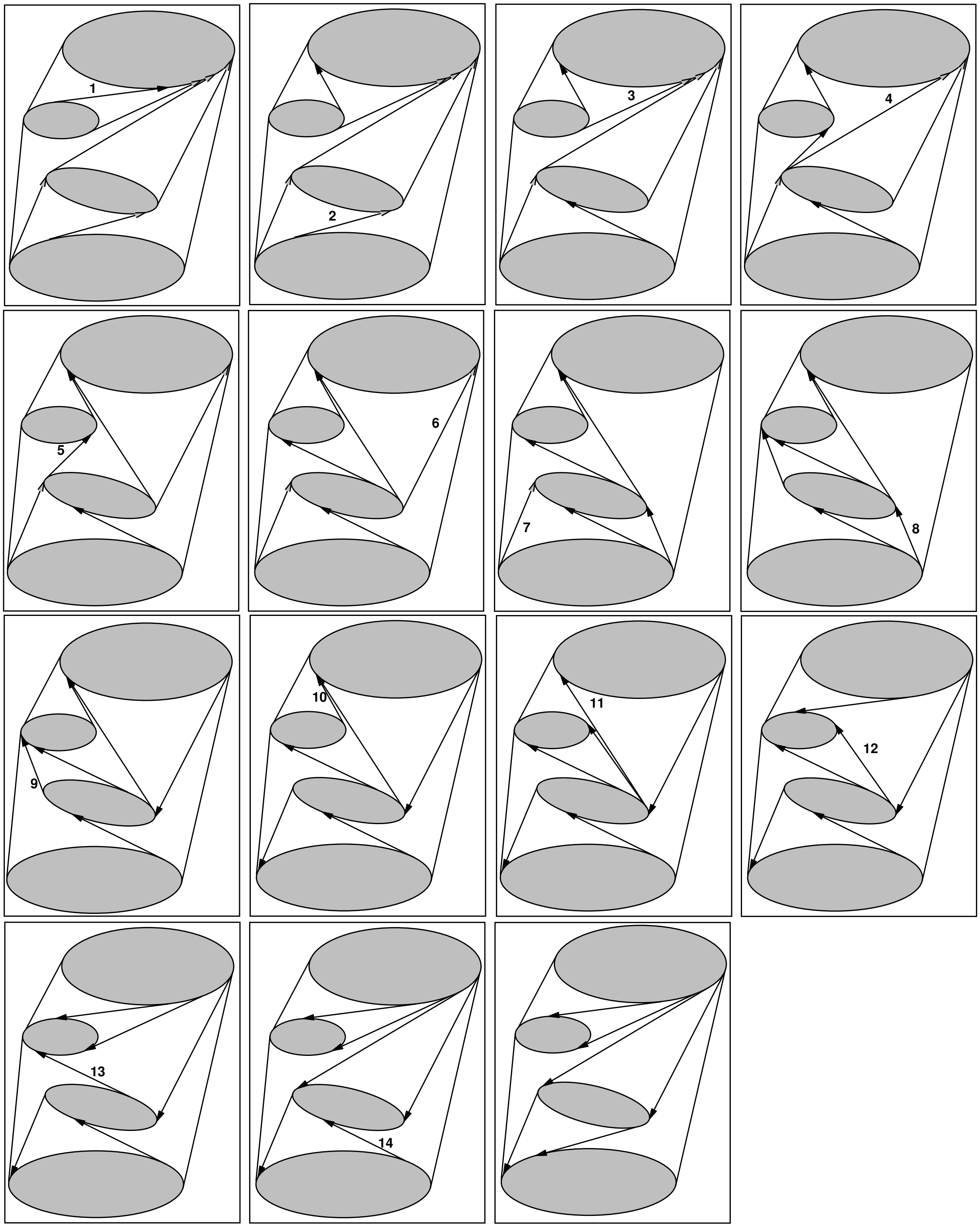


Figure 18: The greedy flip algorithm. In this example at each step the internal bitangent of minimal slope in the current pseudo-triangulation is flipped.