Discrete Comput Geom 16:419–453 (1996)

Discrete & Computational
Geometry

Topologically Sweeping Visibility Complexes via Pseudotriangulations*

M. Pocchiola[1] and G. Vegter[2]

[1] Département de Mathématiques et Informatique, Ecole normale supérieure,
ura 1327 du CNRS, 45 rue d'Ulm, 75230 Paris Cedex 05, France
pocchiola@dmi.ens.fr

[2] Department of Mathematics and Computer Science, University of Groningen,
P.O. Box 800, 9700 AV Groningen, The Netherlands
gert@cs.rug.nl

Abstract. This paper describes a new algorithm for constructing the set of free bitangents of a collection of n disjoint convex obstacles of constant complexity. The algorithm runs in time $O(n \log n + k)$, where k is the output size, and uses $O(n)$ space. While earlier algorithms achieve the same optimal running time, this is the first optimal algorithm that uses only linear space. The visibility graph or the visibility complex can be computed in the same time and space. The only complicated data structure used by the algorithm is a splittable queue, which can be implemented easily using red–black trees. The algorithm is conceptually very simple, and should therefore be easy to implement and quite fast in practice. The algorithm relies on greedy pseudotriangulations, which are subgraphs of the visibility graph with many nice combinatorial properties. These properties, and thus the correctness of the algorithm, are partially derived from properties of a certain partial order on the faces of the visibility complex.

1. Introduction

Visibility graphs (for polygonal obstacles) were introduced by Lozano-Pérez and Wesley [18] for planning collision-free paths among polyhedral obstacles; in the plane a shortest euclidean path between two points runs via edges of the 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

* A preliminary version of this work appeared in the *Proceedings of the* 11*th Annual ACM Symposium on Computational Geometry*, Vancouver, June 1995, pages 248–257.

Table 1. Optimal time visibility graph algorithms.

Space	Time	Source	Obstacles	Data structures
n	n^2	Edelsbrunner–Guibas [8]	n points	Simple
k	$k + n \log n$ $(k + n)$	Hershberger [12]	Simple polygon with n vertices	Simple (Finger search trees + $O(n)$-triangulation algorithm)
k	$k + n \log n$	Ghosh–Mount [10]	n line segments	Split-find data structure of Gabow and Tarjan [9]
k	$k + n \log n$	Pocchiola–Vegter [24]	n convex sets	Split-find
n	$k + n \log n$	This paper	n convex sets	Splittable queues

[4], [8], [10]–[12], [14], [23]–[25], [27], [29], [30] as well as their characterization (see [1], [2], [6], [22], [26], and the references cited therein).

This paper describes a new algorithm for constructing the (tangent) visibility graph of a collection $\mathcal{O}$ of n disjoint convex obstacles of constant complexity. Its running time is $O(n \log n + k)$, where k is the output size, and its working space is linear. The algorithm is extendible to the case where the objects are allowed to touch each other. Therefore, our method can be adapted to compute the (classical) visibility graph of a set of disjoint polygons in the plane (e.g., by triangulating the polygons and applying the extended version of our algorithm to the collection of edges of the triangulation). While earlier algorithms [10], [12], [14], [24] achieve the same optimal running time, under various assumptions on the nature of the obstacles (see Table 1), this is the first optimal algorithm that uses only linear space. The only complicated data structure used by the algorithm is a splittable queue, which can be implemented easily using red–black trees. The algorithm is conceptually very simple, and should therefore be easy to implement and quite fast in practice. We are convinced that the algorithm also works for obstacles of nonconstant complexity; see Section 3.4.5.

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* (resp. *interior*) bitangent is a bitangent lying on the boundary of (resp. in the interior of) the convex hull of the collection of obstacles. We denote by B the set of free bitangents of the collection 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, as illustrated in Fig. 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.*

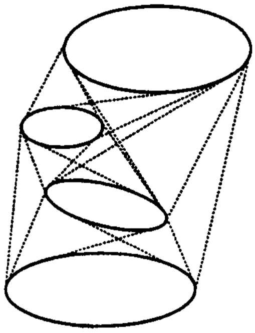

Fig. 1. The visibility graph of a collection of four obstacles.

Our approach is to turn B into a poset (partially ordered set) $(B, \preceq)$ and to compute a linear extension of $(B, \preceq)$, i.e., to embed $\preceq$ into a linear (total) order. In other words, we solve the topological sorting problem [15], [16] for $(B, \preceq)$.

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^{or} be the oriented version (double cover) of B, obtained by associating with each $b \in B$ the two *directed* versions of b. The subset X_0 of $B^{\text{or}} \times \mathcal{R}$ is defined by

$$X_0 = \{(v, u) \in B^{\text{or}} \times \mathcal{R} \mid \exp(u) \text{ is the unit vector along } v\}.$$

A point $b = (v, u)$ in X_0 is called a *bitangent* in X_0; the unoriented version of the bitangent $v \in B^{\text{or}}$ is denoted by $\text{bit}(b)$; $u \in \mathcal{R}$ is called the *slope* of b, denoted by $\text{Slope}(b)$. We identify a bitangent in B with the corresponding bitangent in X_0 with slope in $[0, \pi)$. Two bitangents b and b' in X_0 are *crossing*, *disjoint*, etc., if the corresponding bitangents $\text{bit}(b)$ and $\text{bit}(b')$ in B are crossing, disjoint, etc., as subsets of the plane.

The (partial) order $\preceq$ on X_0 is defined as follows: $b \preceq b'$ if there is a counterclockwise oriented curve joining (some point of) $\text{bit}(b)$ to $\text{bit}(b')$, that runs along the edges (arcs and bitangents) of the visibility graph of the obstacles, and that sweeps an angle of $\text{Slope}(b') - \text{Slope}(b)$, as illustrated in Fig. 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 $\preceq$ (see Lemma 7). Since $\preceq$ is compatible with the slope order on X_0, an obvious extension of $\preceq$ 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 $\preceq$ 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

[1] A filter I of a poset $(P, \preceq)$ is a subset of P such that if $x \in I$ and $x \preceq y$, then $y \in I$. The set of filters, ordered by reverse inclusion, is a poset. Our main interest in the notion of filter is that, given two filters I and J with $J \subseteq I$ and $I \backslash J$ finite, the sequence $x_1, x_2, \ldots, x_k$ of elements of $I \backslash J$ is a linear extension of $(I \backslash J, \preceq)$ if and only if the sequence of sets $I_1, I_2, \ldots, I_k$, defined by $I_i \backslash J = \{x_i, x_{i+1}, \ldots, x_k\}$, is an unrefinable chain of filters in the interval $[I, J]$. We borrow poset terminology from Stanley [28, Chapter 3] and McMullen [19]. To keep the paper self-contained, we review this terminology in Appendix A.

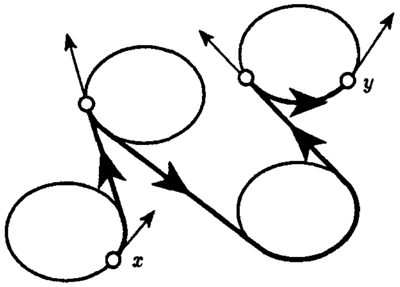

Fig. 2. A counterclockwise oriented curve with initial point x, terminal point y, and two cusp points. The two cusp points subdivide the curve into three regular smooth counterclockwise subcurves, i.e., each of these three subcurves can be described by a function f: $[0, 1] \to \mathcal{R}^2$ with $f'(t) = |f'(t)| \exp(\theta(t)) \neq 0$, where $\theta(t)$ is a continuous nondecreasing function. By definition the angle swept by the regular subcurve f is $\theta(1) - \theta(0)$, and the angle swept by the curve is the sum of the angles swept by its regular subcurves. In the example the angle swept between the two rays starting at x and y is slightly over 2π.

$I(u)$, 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 $(X_0, \preceq)$ we define a maximal subset $G(I) = \{b_1, \ldots, b_m\}$ 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 $\min_{\preceq} I \subseteq G(I)$. We prove that for each filter I the set $G(I)$ contains $3n - 3$ bitangents, that subdivide free space into regions called *pseudotriangles.* This subdivision, also denoted by $G(I)$, is called a greedy pseudotriangulation. The regions owe their name to their special shape, that is explained in more detail in Section 2. We refer to Fig. 3 for an example of greedy pseudotriangulations associated with filters of X_0.

Our algorithm maintains the greedy pseudotriangulation $G(I)$ as I ranges over a maximal chain of filters of the interval $[I(0), I(\pi)]$, namely the set of filters I with $I(0) \supseteq I \supseteq I(\pi)$. The basic operation that updates the pseudotriangulation 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 $(X_0, \preceq)$ and let $b \in \min_{\preceq} I$. Then $G(I\backslash\{b\})$ is obtained from $G(I)$ by flipping b, i.e., by replacing b with the only minimal bitangent in $I\backslash\{b\}$ disjoint from the other bitangents in $G(I)$ (see Fig.* 3).

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 [8]. Our pseudotriangulations replace their (upper and lower) horizon trees.

The paper is organized as follows. In Section 2 we recall the definition of the visibility complex Y of the collection of obstacles, a cell complex on the space $\mathcal{S}^2 \times \mathcal{S}^1$ carrying the view from points (in $\mathcal{S}^2$, the plane $\mathcal{R}^2$ together with the point at infinity) along a direction (in $\mathcal{S}^1$).

The set X_0, introduced in this section, is the set of vertices (0-faces) of the universal

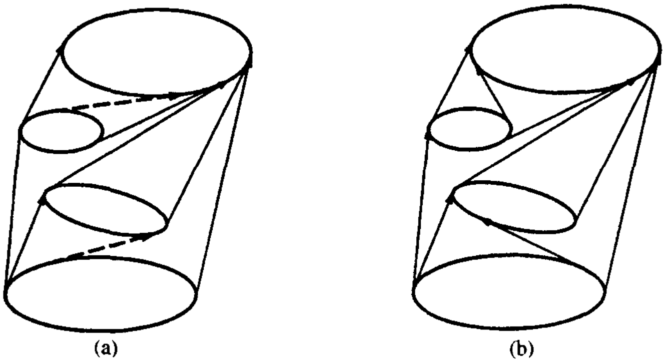

Fig. 3. (a) The greedy pseudotriangulation associated with the filter $I(0)$ of bitangents with slope ≥ 0. The dashed bitangents b_1 and b_2 are both minimal in the filter $I(0)$. (b) The greedy pseudotriangulation associated with the filter $I(0)\backslash\{b_1, b_2\}$ which is obtained from $G(I(0))$ by flipping b_1 and b_2.

cover X of Y, which is a cell complex on the universal cover $\mathcal{S}^2 \times \mathcal{R}$ of $\mathcal{S}^2 \times \mathcal{S}^1$, induced by the universal covering map exp: $\mathcal{R} \rightarrow \mathcal{S}^1$ on the second component. We introduce the partial order $\preceq$ on the cover X and we prove that this order satisfies a "minimum-element" property: the set of bitangents greater than a given bitangent and crossing it has a minimum element. Then we prove Theorem 2 by interpreting the greedy pseudotriangulations as maximal antichains of $\preceq$ on $X\backslash X_0$. In Section 3 we show how the flip operation can be efficiently implemented, using splittable queues.

2. The Visibility Complex

2.1. *Terminology: Pseudotriangles and Pseudotriangulations*

Let $\mathcal{O} = \{O_1, O_2, \dots, O_n\}$ be a collection of n pairwise disjoint closed convex sets O_i (obstacles for short). We assume that the obstacles are (1) *strictly* convex (i.e., the open line segment joining two points of an obstacle lies in its interior), (2) *smooth* (i.e., there is a well-defined tangent line through each boundary point), and (3) in *general position* (i.e., no three obstacles share a common tangent line). In particular two bitangents in B are disjoint or intersect transversally (i.e., not at their endpoints). These assumptions are only for ease of exposition. The general case can be treated by standard perturbation techniques; for example, to cover the case where obstacles are allowed to be points and disjoint line segments the perturbation scheme may, e.g., consist of taking the Minkowski sum with an infinitesimally small circle. A *pseudotriangulation* 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 pseudotriangulation always exists and that the bitangents of the boundary of the convex hull of the obstacles are edges of any pseudotriangulation. Two pseudotriangulations of a collection of four obstacles are depicted in Fig. 3. The subdivision owes its name to the special shape of its regions. A

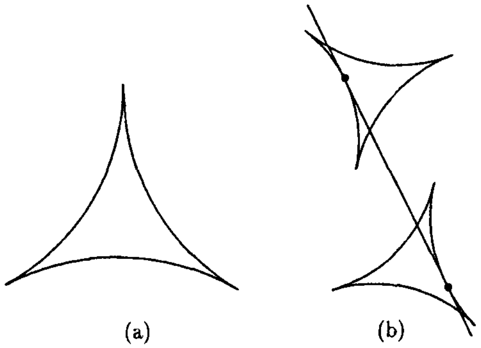

Fig. 4. (a) A pseudotriangle. (b) Two disjoint pseudotriangles share exactly one common tangent line.

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. See Fig. 4(a). These three endpoints are 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 $\varphi_T \colon \mathcal{S}^1 \to \mathcal{R}$ of T is well defined, continuous, and satisfies $\varphi_T(u) = -\varphi_T(-u)$).

Lemma 3. *The bounded free regions of any pseudotriangulation are pseudotriangles. Furthermore, the number of pseudotriangles* (*of a pseudotriangulation 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 free 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 this we derive that R is not maximal. This means that this face is not simply connected or that its exterior boundary contains at least four cusp points. In both cases we add to R a bitangent as follows. Take the minimal length closed curve, homotopy equivalent to the exterior boundary of the face, and going through all cusp points of the exterior boundary but one. This closed 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 one 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

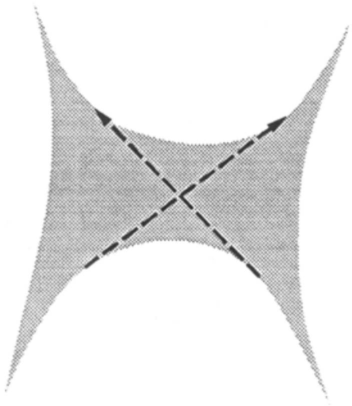

Fig. 5. A pseudoquadrangle and its diagonals.

vertices. Finally, the total number of bounded regions is equal to the sum of the number of pseudotriangles and the number (n) of obstacles. □

Lemma 4. *Let T and T' be two disjoint pseudotriangles. Then T and T' have exactly one common tangent line.* (*See Fig.* 4(b).)

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 uniqueness we observe that tangent lines to a pseudotriangle cross inside the pseudotriangle. □

We use this last lemma only in the case where T and T' are adjacent pseudotriangles (in a pseudotriangulation). 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 Fig. 5).

2.2. *Definition of the Visibility Complex Revisited*

The visibility complex was defined in [24] as a partition of the set of free rays. Here we define the visibility complex as a partition of the whole set of rays (free or not free) augmented with rays at infinity. This slight modification simplifies the description of the combinatorial structure of the visibility complex and, in particular, of its cross sections.

We 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 $u \in \mathcal{R}$ let C_u be an infinite strip, centered around a line through the origin with slope $u + \pi/2$, large enough to contain all the obstacles. We denote by L_u and R_u the two connected components of $\mathcal{R}^2 \backslash C_u$, where L_u comes before R_u along

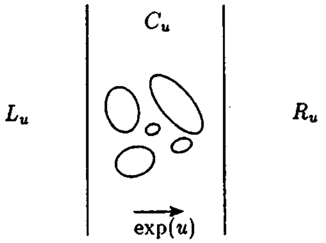

Fig. 6. The infinite strip C_u.

lines with direction exp(u) (see Fig. 6). The u-free space F_u is the closure of $C_u \setminus \bigcup \mathcal{O}$. 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} (resp. 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 (resp. right) half-plane; obviously γ_{+i} and γ_{-i} are homeomorphic to $\mathcal{R}$. Similarly, we denote by $\gamma_{+\mathcal{O}}$ (resp. $\gamma_{-\mathcal{O}}$) the set of rays (p, u) emanating from and tangent to the convex hull of the set of obstacles.

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, $(p, u + \pi)$). By definition the backward (resp. forward) view from the point at infinity along u is L_u (resp. R_u). The view from a point p inside an object O_i is this object O_i, irrespective of the direction.

We define a cell complex X, whose underlying space $|X|$ is a quotient space of the space of rays $\mathcal{S}^2 \times \mathcal{R}$. More precisely, for $p, q \in \mathcal{S}^2$ and $u \in \mathcal{R}$, with $p \neq q$, we declare (p, u) equivalent to (q, u) iff (1) the slope of the directed line from p to q is equal to u, up to an integer multiple of π, and (2) the line segment $[p, q]$ lies in u-free space F_u. In this situation we write $(p, u) \sim (q, u)$. The space $|X|$ is the quotient space of $\mathcal{S}^2 \times \mathcal{R}$ under the reflexive, transitive closure of $\sim$. By a slight abuse of terminology, an equivalence class is called a *ray* in $|X|$. 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 refer to this set as the *cross section* of $|X|$ at u.

If p is a point in u-free space, the equivalence class of (p, u) consists of all points of the form (q, u), where the point q ranges over the largest line segment with slope u in F_u, that contains p. One may think of the cross section of $|X|$ at u as obtained from $\mathcal{S}^2$ by contracting[2] the latter line segment, for all points p in free space. The reader may find it helpful to refer to the top half of Fig. 11. The rightmost part of that figure contains a schematic picture of the cross section of $|X|$ at $u = 0$ (provided we forget about the direction of the labeled edges). The labeled edges can be seen to represent equivalence classes under $\sim$, defined for the set of obstacles in the leftmost part of the figure (the

[2] We refer to the video segment [7] for an illustration of this contraction process.

sets L_0 and R_0 are not depicted here). The subset F_0 is subdivided into a number of strip-shaped regions. Each point on a labeled edge of the cross section corresponds to a maximal free horizontal line segment in the strip-shaped region having the same label. Also note that points inside an object constitute an equivalence class by themselves, giving rise to the two-dimensional regions in the cross section. It is not hard to see that the cross section is homeomorphic to $\mathcal{S}^2$. In Fig. 11 the edge numbered 16 continues directly into the edge numbered 1 via the point at infinity.

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). 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. 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 $i \in \{1, \dots, n, +, -\}$, 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 three-dimensional cell (or face) decomposition of $|X|$, denoted 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 $i \in \{1, \dots, n, +, -\}$. 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 collections of rays which are tangent to two obstacles. A face x is said to be *bounded* if 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 $\mathcal{S}^2$. We denote the sets of 0-, 1-, 2-, and 3-faces of X by X_0, X_1, X_2, and X_3, respectively, and the set of bounded 2-faces by X_2^*.

Let π be the mapping which associates the ray $(p, u+\pi)$ with the ray (p, u). Clearly, the (induced) mapping $\pi\colon |X| \to |X|$ is an automorphism of the complex X. The quotient complex $Y := X/\pi^2$ (whose underlying space is now $\mathcal{S}^2 \times \mathcal{S}^1$) is the *visibility complex* of the collection of obstacles. (In [24] the visibility complex was defined as the 2-skeleton of X/π^2.)

Let $P(X)$ be the poset of faces of X, augmented with $\emptyset$ and $|X|$, ordered by the inclusion relation of their closures. Similarly, we define $P(Y)$ to be the poset of faces of Y. The local combinatorial structure of $P(X)$ or $P(Y)$ is described in the following theorem. (See Fig. 7 and also [24]. We refer to Appendix A for the terminology on abstract polytopes.)

Theorem 5. *$P(X)$ $(P(Y))$ is an abstract polytope of rank 4. Furthermore, the vertex-figure of a vertex is the face poset of a three-dimensional simplex.*

Note that there is a canonical mapping *arc* from the set X_1 of edges of X onto the set of arcs on the boundaries of the obstacles (these arcs correspond to edges of the visibility graph of $\mathcal{O}$, see Section 1). More precisely, for $x \in X_1$, the arc arc(x) consists of the origins of the rays in x emanating from the object to which they are tangent. The canonical mapping from the set X_0 of vertices of X onto the set B of free bitangents of $\mathcal{O}$ is denoted by *bit*; see Fig. 7. In particular, the preimage under the mapping bit of the

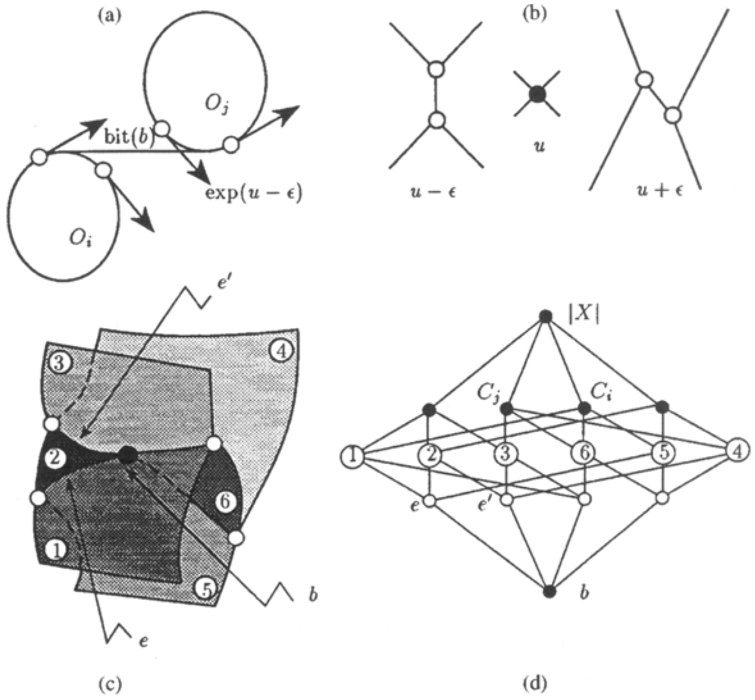

Fig. 7. (a) Two obstacles defining a vertex b of the visibility complex with slope u. (b) (Local) cross sections at slopes $u - \varepsilon$, u, and $u + \varepsilon$. (c) Neighborhood of a vertex of the visibility complex. (d) The Hasse diagram of the vertex-figure of a vertex of $P(X)$

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 is called a *bitangent* in X_0.

A *pseudotriangulation* 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) bit(G) is a pseudotriangulation of the collection of obstacles, and (2) Card $G = 3n - 3$.

Let x be a 1-face (namely an edge) or a bounded 2-face in X. We define $\sup x$ (resp. $\inf x$) to be the ray with maximal (resp. minimal) slope in the closure of x. The operator sup (resp. inf) is a one-to-one correspondence between the set of bounded 2-faces in X_2 and the set of vertices in X_0. For a vertex x we denote by $\sup(x)$ the unique 2-face y with $\inf(y) = x$. Similarly, $\inf(x)$ is the unique 2-face y with $\sup(y) = x$. In this way inf and sup are defined for all vertices, edges, and bounded 2-faces of X.

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 that are the canonical images of the curves of rays $\gamma_{+\mathcal{O}}$ and $\gamma_{-\mathcal{O}}$.

Remark 6. The numbers of 0-, 1-, 2-, and 3-faces of the visibility complex Y are $2k, 4k, 2k + 1$, and $n + 2$, respectively; here k is the number of free bitangents. This equality is a consequence of the previous discussion, namely 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 $P(Y)$ is 24 times the number of vertices, i.e., $48k$.

2.3. *The Poset $(X, \preceq)$ and the "Minimum-Element" Property*

Now we turn X into a poset $(X, \preceq)$ by taking the transitive and reflexive closure of the relation $\preceq$, defined by

$$\inf x \preceq x \preceq \sup x, \qquad \forall x \in X_1 \cup X_2^*, \tag{1}$$

i.e., for $t, t' \in X_0$ we have $t \preceq t'$ 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 $x_1, \ldots, x_l$ by the sequence of edges of its upper (or lower) boundary. In other words, $t \preceq t'$ 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, $\preceq$ is compatible on X_0 with the slope order. Observe that for all $x \in X_1 \cup X_2^*$ the cell $\sup x$ (resp. x) covers the cell x (resp. $\inf x$). Finally note that the unbounded cells are isolated points in $(X, \preceq)$.

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

Lemma 7. *Let t and t' be two bitangents in X_0.*

(1) *If* $\mathrm{bit}(t)$ *and* $\mathrm{seg}(t')$ *are crossing, then t and t' are comparable with respect to $\preceq$.*
(2) *If* $\mathrm{seg}(t)\backslash\mathrm{bit}(t)$ *and* $\mathrm{seg}(t')\backslash\mathrm{bit}(t')$ *are crossing, say in point p, and if there is no free line segment emanating from p, tangent to an obstacle in $\mathcal{O}$, and lying in the wedge $t_+\backslash t'_+$ (here t_+ is the open half-plane bounded by the supporting line of* $\mathrm{bit}(t)$*, that contains the line segment* $\mathrm{bit}(t')$*), then t and t' are comparable with respect to $\preceq$.*
(3) *$t \preceq \pi^k(t')$, for all sufficiently large k.*

Proof. Assume first that $\mathrm{bit}(t)$ and $\mathrm{bit}(t')$ are crossing. Clearly it suffices to prove that $\mathrm{bit}(t)$ and $\mathrm{bit}(t')$ are the diagonals of a pseudoquadrangle of some pseudotriangulation. To show the existence of a such pseudotriangulation we add four sufficiently small obstacles near the crossing point of $\mathrm{bit}(t)$ and $\mathrm{bit}(t')$ as indicated in Fig. 8(a). Now we consider a pseudotriangulation (of the set of $n+4$ obstacles) that contains the bitangent $\mathrm{bit}(t)$, and the $3 \times 4 = 12$ bitangents shown dashed in Fig. 8(a). Up to some flip operations we can assume that these 12 bitangents are the only bitangents that emanate from the 4 new obstacles. Removing these 4 added obstacles and their 12 bitangents

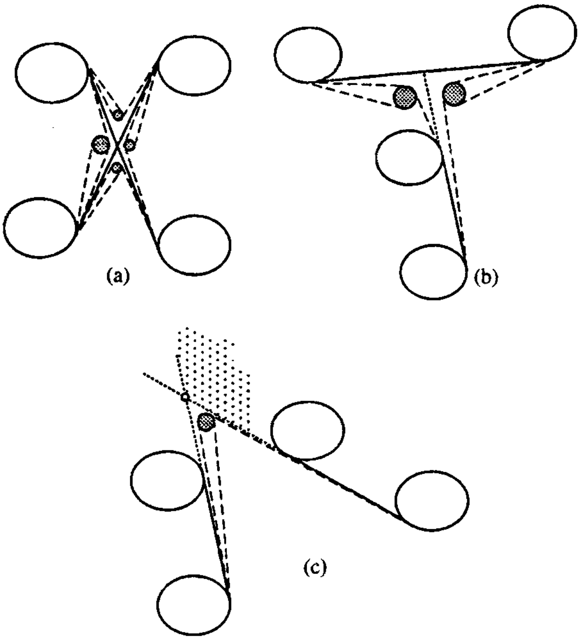

Fig. 8. (a) bit(t) and bit(t') are crossing. The 4 added obstacles and the 12 added bitangents are shown dashed. (b) seg(t)\bit(t) and bit(t') are crossing. (c) seg(t)\bit(t) and seg(t')\bit(t') are crossing.

yields a pseudotriangulation (since the number of remaining bitangents is $3n - 3$) with the desired property. A similar construction yields the result in the case where bit(t) and bit(t') are disjoint, either in the case where seg(t)\bit(t) and bit(t') are crossing (see Fig. 8(b)), or in the case where seg(t)\bit(t) and seg(t')\bit(t') are crossing (see Fig. 8(c)). In this latter case the condition given in the lemma ensures that up to some flip operations the added obstacle contributes only to the three dashed bitangents. After removing the added obstacles and their bitangents, bit(t) and bit(t') are edges of the same pseudotriangle, and hence they are comparable.

Now we prove claim (3). We can assume that bit(t) and bit(t') are disjoint. Consider a pseudotriangulation G that contains bit(t) and bit(t'), and consider a curve in free space that joins bit(t) and 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 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 $t_j \preceq \pi^{k_j}(t_{j+1})$ for k_j sufficiently large. It follows that $t \preceq \pi^k(t')$ for k sufficiently large ($k = \sum_j k_j$). □

Now we come to the "minimum-element" property announced in the Introduction. We denote by φ the one-to-one mapping

$$t \in X_0 \mapsto \sup\sup t \in X_0, \tag{2}$$

i.e., $\varphi(t)$ is the ray with maximal slope in the (closure of the) face for which t is the ray with minimal slope. It can easily be checked that $\varphi \circ \pi = \pi \circ \varphi$.

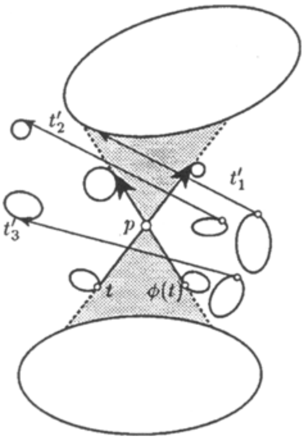

Fig. 9. Illustration of the proof of the "minimum-element" property.

Lemma 8 ("Minimum-Element" 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 $\varphi(t) \preceq t'$ (and $t \preceq \varphi^{-1}(t')$). In other words, $\varphi(t)$ is the minimum bitangent in the set of bitangents crossing t and larger than t.*

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

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

Clearly, T is a subset of free space. Therefore the slope of t' is greater than the slope of $\varphi(t)$, and bit$(\varphi(t))$ and seg(t') are crossing (first case), or bit(t') is tangent to the boundary of T (second case). See Fig. 9 for an illustration. Hence it suffices to prove that t' and $\varphi(t)$ are comparable with respect to $\preceq$ in order to conclude that $\varphi(t) \preceq t'$. The first case is covered by Lemma 7, claim (1). In the second case 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 7. □

Remark 9. 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; this set has a minimum element, namely $\pi(t)$.

2.4. *Filters, Antichains, and Greedy Pseudotriangulations*

For a finite subset A of X we define the filter A^+ of $(X_0, \preceq)$ by

$$A^+ = \{x \in X_0 \mid y \preceq x \text{ for some } y \in A\}. \tag{3}$$

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

$$\hat{I} = \{x \in X_1 \cup X_2^* \mid \sup x \in I,\ \inf x \notin I\} \cup \{\text{unbounded faces}\}. \tag{4}$$

Note that, by definition, the set of unbounded faces is a subset of $\hat{I}$.

A *proper* filter of a poset $(X, \preceq)$ is a filter which is a nonempty proper subset of X.

Lemma 10. *The mapping $I \mapsto \hat{I}$ is a one-to-one correspondence between the set of proper filters of $(X_0, \preceq)$ and the set of maximal antichains of $(X \setminus X_0, \preceq)$, whose inverse is the map $A \mapsto A^+$.*

Proof. First we show that $\hat{I}$ is a maximal antichain of $(X \backslash X_0, \preceq)$. Let $x \in \hat{I}$, $y \in X \backslash X_0$ 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 $x \in X \backslash X_0$ and consider the unrefinable chain $\{\ldots, \inf^2(x), \inf(x), x, \sup(x), \sup^2(x), \sup^3(x), \ldots\}$. By Lemma 7, part (3), this chain joins $X_0 \backslash I$ and I. Consequently, this chain intersects $\hat{I}$, and x is comparable with an element in $\hat{I}$. Finally observe that $(\hat{I})^+ = I$, since (1) $\min I \subset (\hat{I})^+$ and (2) $(\min I)^+ = I$. Note that, in view of Lemma 7, part (3), I contains no infinite decreasing chains. □

Theorem 11. *Let A be a maximal antichain in $(X \backslash X_0, \preceq)$. 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, $P(A)$, the subposet of $P(X)$ induced by A, is an abstract polytope of rank 3.*
2. *The numbers of 1-, 2-, 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) and Theorem 5, the number of incidences between edges and 2-faces of a maximal antichain is three times the number of edges, and twice the number of 2-faces. Therefore the number of 2-faces is $3n$. Planarity is proved by computing the Euler characteristic. □

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 noncomparable they are pairwise disjoint, and consequently $\bigcup_{i\geq 1} B_i(I)$ is a pseudotriangulation in X (in particular $B_i(I) = \emptyset$ for i sufficiently large). This pseudotriangulation is denoted by $G(I)$ and is called the *greedy* pseudotriangulation associated with the filter I.[3] Finally, for a filter I we define

$$S(I) = \{b \in I \mid \varphi^{-1}(b) \notin I\}. \tag{5}$$

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 12.

(1) *For all filters I, and all interior (resp. exterior) bitangents $b \in \min I$, the set difference $G(I\backslash\{b\})\backslash G(I)$ is equal to $\{\varphi(b)\}$ (resp. $\{\pi(b)\}$).*
(2) *For all filters I, all bitangents $b \in G(I)$, and all $t \in I$ crossing b, we have $b \preceq t$.*
(3) *For all filters I we have $G_{\text{int}}(I) = S_{\text{int}}(I)$.*

Proof. Claims (1) and (2) are obvious in the case where b is an exterior bitangent (see Remark 2); therefore we assume now that b is an interior bitangent. We prove the theorem by showing that claim (3) implies (1), and subsequently that (1) implies (2), after which we establish the truth of claim (3).

First observe that $\varphi(b)$ is disjoint from any $b' \in G(I)\backslash\{b\}$, otherwise $\varphi(b)$ and b' are comparable, with $b' \prec \varphi(b)$ (indeed if $\varphi(b) \prec b'$, then, according to Lemma 8, $\varphi(b) \preceq \varphi^{-1}(b')$; consequently $\varphi^{-1}(b') \in I$, i.e., $b' \notin G(I)$). According to Lemma 8 this implies that $b' \preceq b$, a contradiction with $b \in \min I$. Therefore, it is sufficient to prove that $\varphi(b)$ is a bitangent in $G(I\backslash\{b\})$. Suppose the contrary holds. Then $\varphi(b)$ intersects some $b' \in G(I\backslash\{b\})$, with $b' \prec \varphi(b)$. However, according to Lemma 8, this implies that $b' \preceq b$, a contradiction. Thus, claim (3) implies claim (1).

Now we prove that claim (1) implies claim (2). To this end let I be a filter, let b be a bitangent in $G(I)$ and let t be a bitangent in I which crosses b. We define the sequence of filters $I_1, I_2, \ldots$ by $I_1 = I$ and $I_{k+1} = I_k\backslash B_1(I_k)$. Observe that if $b \in G(I_k)\backslash B_1(I_k)$ 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. Thus, claim (1) implies claim (2).

Finally we prove claim (3) by proving successively that:

(i) $S_{\text{int}}(I) \subseteq G_{\text{int}}(I)$ (in particular the bitangents in $S_{\text{int}}(I)$ are pairwise disjoint).
(ii) $G_{\text{ext}}(I) \subset S_{\text{ext}}(I)$ and Card $S_{\text{ext}}(I) =$ Card $G_{\text{ext}}(I) + 2$.
(iii) Card $S(I) = 3n - 1$.

These three properties imply that $G_{\text{int}}(I) = S_{\text{int}}(I)$, since Card $G(I) = 3n - 3$.

[3] Observe that if $\preceq_1$ is a total order on I, compatible with $\preceq$ on I, then the elements of the set $G(I)$ can be enumerated as the sequence $b_1, b_2, \ldots, b_{3n-3}$, where (1) b_1 is the minimum bitangent in $(I, \preceq_1)$, and (2) b_{i+1} is the minimum bitangent in $(I, \preceq_1)$ disjoint from $b_1, b_2, \ldots, b_i$.

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 we have $b \preceq b'$. In the first case Lemma 8 implies that $b' \preceq \varphi^{-1}(b)$, 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).

For an exterior bitangent t lying on $\gamma_{+\mathcal{O}}$ (resp. $\gamma_{-\mathcal{O}}$) we denote by $succ(t)$ the minimal exterior bitangent greater than t lying on $\gamma_{+\mathcal{O}}$ (resp. $\gamma_{-\mathcal{O}}$). Observe that $succ(\pi(t)) = \pi(succ(t)) = \varphi(t)$ whenever t is an exterior bitangent 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 $\gamma_{+\mathcal{O}}$. Since $succ^{-1}(t) \notin I$ it follows that $\varphi^{-1}(t)$ and $\varphi^{-1}(\pi(t))$, which are respectively equal to $\pi^{-1}(succ^{-1}(t))$ and $succ^{-1}(t)$, are not in I; thus t and $\pi(t)$ are both in $S(I)$. A similar result holds for the minimal element in I, say t', lying on $\gamma_{-\mathcal{O}}$. 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 $\gamma_{+\mathcal{O}}$. Hence, $succ^k(t) = \pi(t')$, since $\pi(t') \in S(I)$ and $succ(\pi(t')) = \varphi(t') \notin S(I)$. Similarly, $succ^{k'}(t') = \pi(t)$, where k' is the greatest 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 $G_{\text{ext}}(I)$ 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, $G_{\text{ext}}(I) \subseteq S_{\text{ext}}(I)$, since $\pi(t)$ and $\pi(t')$ are not in $G_{\text{ext}}(I)$. Furthermore, a cardinality argument shows that $S_{\text{ext}}(I) = G_{\text{ext}}(I) \cup \{\pi(t), \pi(t')\}$. This proves claim (ii).

Finally note that $S(I) = \sup \hat{I} = \sup \hat{I} \backslash X_1$, and consequently Card $S(I) = 3n - 1$, according to Theorem 11. This proves claim (iii), and therefore completes the proof of the theorem. □

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 at least u. Theorem 2 suggests a very simple algorithm: maintain the greedy pseudotriangulation $G(I)$, while I ranges over a maximal chain of filters in the interval $[I(0), I(\pi)]$.

Algorithm GREEDY FLIP ALGORITHM

```
compute the greedy pseudotriangulation $G := G(I(0))$;
repeat
    select a minimal bitangent $b$ in $G$ with slope less than $\pi$;
    flip $b$;   (i.e., replace $b$ by $\varphi(b)$ (resp. $\pi(b)$) if $b$ is an interior (resp. exterior)
        bitangent)
until there are no more bitangents of slope less than $\pi$.
```

Theorem 2 proves the correctness of this algorithm. Of course, we still have to explain how to implement the flip operation (namely line 4) and how to select a minimal bitangent with slope less than π (namely line 3), so that the total cost of these operations is $O(k)$ time. Figure 10 illustrates the greedy flip algorithm. In this example the flipped bitangent has minimal slope, and is therefore a minimal element with respect to the partial order $\preceq$.

In Section 3.2 the construction of the initial pseudotriangulation $G(I(0))$ 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 Pseudotriangulation $G(I(0))$*

Lemma 13. *The greedy pseudotriangulation $G(I(0))$ 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. For simplicity assume that no free bitangent has slope 0. A useful aid in the construction of $G := G(I(0))$ 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 $O(u, \textit{left})$ and $O(u, \textit{right})$. The collection of all these points is denoted by $V(u)$.

Two distinct objects O and O' have exactly eight common directed tangent lines. They form four pairs, denoted by (O, O', τ, τ'), where τ and τ' are either *left* or *right*. For instance, $(O, O', \textit{left}, \textit{right})$ is the tangent line going from O to 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 $u + \pi$. 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 11 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

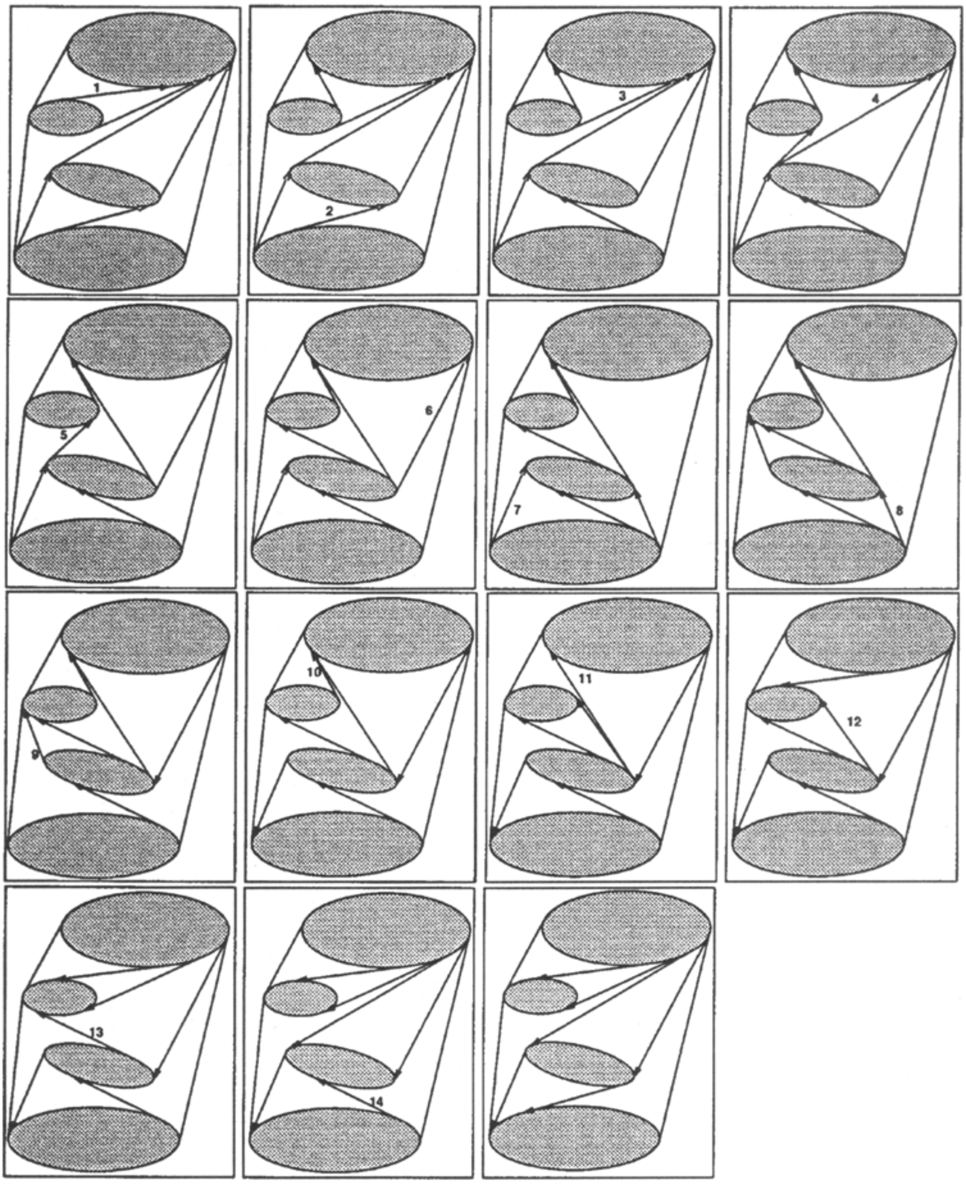

Fig. 10. The greedy flip algorithm. At each step the internal bitangent of minimal slope in the current pseudotriangulation is flipped.

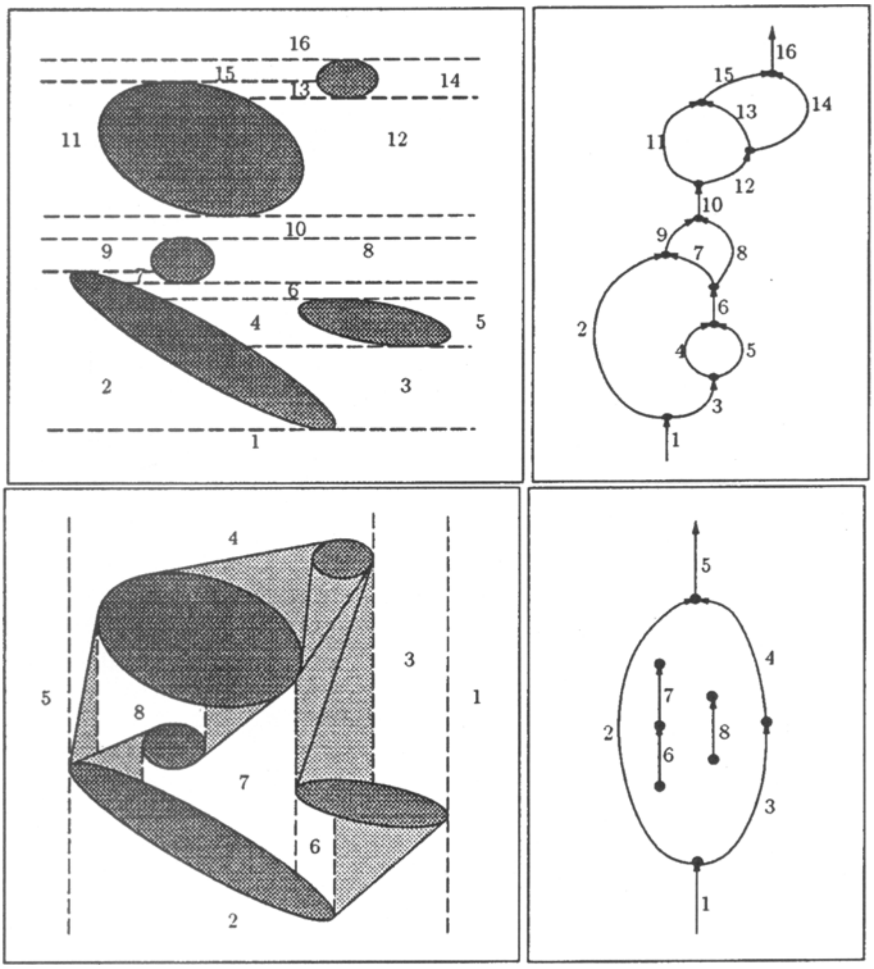

Fig. 11. (a) The labeled regions in the upper left part form the faces of the initial visibility map $M(0)$. The graph $\Gamma(0)$ is depicted in the upper right part. (b) The labeled regions in the lower left part, together with the lightly shaded regions, are the regions of the visibility map $M(\pi/2)$. The lower right part represents the graph $\Gamma(\pi/2)$.

standard sweep line algorithm. Furthermore, the subdivision $M(\pi)$ is just the greedy pseudotriangulation G (if we forget about the four unbounded faces that partition the complement of the convex hull). So we try to maintain $M(u)$ as u ranges over $[0, \pi]$.

We describe the construction of the sequence $B(\pi)$ of bitangents belonging to the pseudotriangulation. This method can be extended in a straightforward way to maintain $M(u)$ as well. The appearance of a free bitangent corresponds to the disappearance of a quadrangular region. For example, in the situation depicted in the lower left part of Fig. 11 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

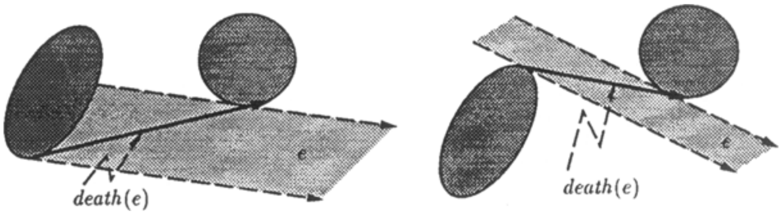

Fig. 12. *death*(e) is the critical direction associated with region e.

the subdivision corresponding to the *quadrangular* regions of $M(u)$ by a directed graph $\Gamma(u)$, 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 $\Gamma(u)$, whose set of edges is in one-to-one 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 Fig. 11. There are two infinite edges, corresponding to the quadrangular faces that contain only one point of $V(u)$ in their boundary. The graph $\Gamma(0)$ contains $3n + 1$ edges, and $\Gamma(\pi)$ contains four 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. This is of course related to Lemma 3.

Consider now an edge e of the graph $\Gamma(u)$. Its terminal points 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 Fig. 12.

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

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

The following obvious result is crucial for the correctness of the algorithm constructing the initial pseudotriangulation.

Lemma 14. *Let the unit vectors u' and u'' be the directions of two consecutive elements of $B(\pi)$.*

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, namely (i) updating the graph $\Gamma(u)$ 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 13 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 Fig. 13. It also holds in the case where at least one of the regions a, b, c, and d is triangular, as illustrated in Fig. 14. Note that the triangular regions *grow* during the

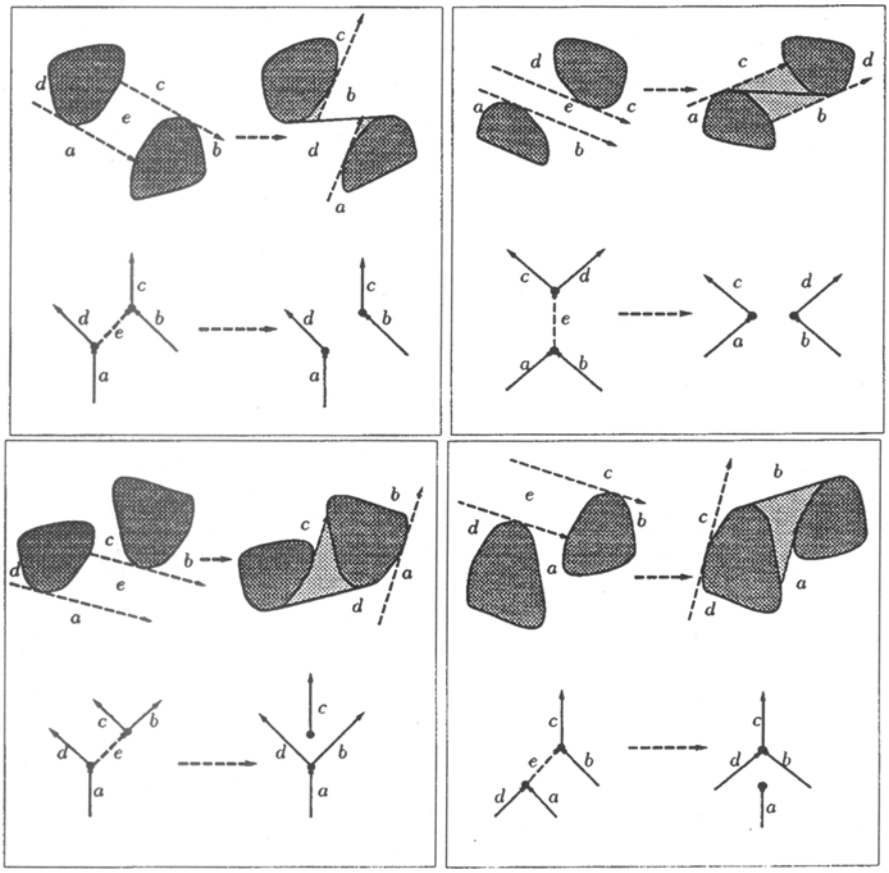

Fig. 13. Transitions during the construction of the pseudotriangulation.

sweep, so not all combinations of triangular and quadrangular regions are possible. For instance, in the upper left part of Fig. 13 it is not possible that region a is triangular whilst at the same time d is quadrangular, since in that case triangular region a does not grow: it shrinks near the edge along which it is incident with d. Finally the pseudotriangulation $G(I(0))$ can easily be computed from the set of bitangents $B(\pi)$.

3.3. *Minimal Bitangents*

Consider a filter I, a bitangent b in the greedy pseudotriangulation $G(I)$, and a pseudotriangle T of $G(I)$. We denote by B_T the set of bitangents $t \in G(I)$ such that 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 by b_T. We denote by *Ltri*(b) (resp. *Rtri*(b)) the pseudotriangle of $G(I)$ incident upon bit(b) and—locally—to the left (resp. right) of bit(b), oriented along the direction of b. The initial point of a directed line segment b

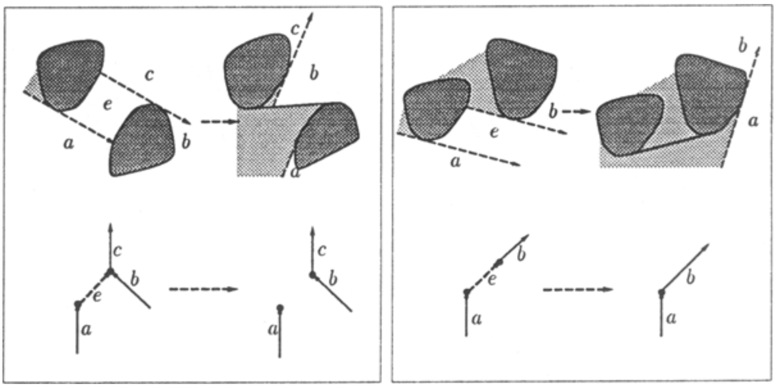

Fig. 14. Transitions during the construction of the pseudotriangulation: at least one of the regions a, b, c, and d is triangular, and hence not represented by an edge in the graph $\Gamma(u)$.

is denoted by *Tail*(b), the terminal point by *Head*(b); so b is directed from its tail to its head. The *basepoint* of T, denoted by p_T, is the tail of b_T, if $T = \mathit{Rtri}(b_T)$, or the head of b_T, if $T = \mathit{Ltri}(b_T)$.

The direction of the tangent line in a point p of ∂T is uniquely determined by the requirement that its slope lies in the interval $[\text{Slope}(b_T), \text{Slope}(b_T) + \pi)$. This slope is also called the *slope* of p. Note that the slope is continuous on ∂T, except at the basepoint of T. A directed subsegment of ∂T is called a *walk* (resp. *reverse walk*) along ∂T if, going from the initial to the terminal point of the subsegment, we pass the points of the subsegment in order of increasing (resp. decreasing) slope. A walk (resp. reverse walk) goes around T clockwise (resp. counterclockwise) when viewed from inside T. In particular, the walk starting at the basepoint of T defines a linear order on the set of bitangents in $\text{bit}(B_T)$, called the *slope order*, which coincides, via the mapping bit, with the linear order $\prec$ on B_T. We denote by b_+ (resp. b_-) the minimal bitangent in $G(I)$ lying on $\gamma_{+\mathcal{O}}$ (resp. $\gamma_{-\mathcal{O}}$), if it exists.

Lemma 15. *Let I be a filter. Then an interior* (*resp. exterior*) *bitangent b is minimal in I if and only if $b = b_{\mathit{Rtri}(b)} = b_{\mathit{Ltri}(b)}$* (*resp. $b = b_{\mathit{Rtri}(b)} = b_-$, or $b = b_{\mathit{Ltri}(b)} = b_+$*).

Proof. Assume b is an interior bitangent. Let e and $e' \in X_1$ be such that $\sup(e) = \sup(e') = b$, and such that $\text{arc}(e)$ and $\text{arc}(e')$ are on the boundaries of *Rtri*(b) and *Ltri*(b), respectively. Clearly, $b = b_{\mathit{Rtri}(b)}$ (resp. $b = b_{\mathit{Ltri}(b)}$) iff $e \in \hat{I}$ (resp. $e' \in \hat{I}$). Since b is minimal in I iff $e, e' \in \hat{I}$, the result follows. A similar argument applies if b is an exterior bitangent. □

The successive cusps we pass during a walk starting at the basepoint of T, are denoted by x_T, y_T and z_T. If the basepoint is a cusp, then by definition it is z_T. The *forward* and *backward* T-views of point p in ∂T are the *points* of intersection of ∂T with the tangent line at p, lying ahead of and behind p, respectively. The point, whose forward (resp. backward) T-view is p_T, if $T = \mathit{Rtri}(b_T)$ (resp. $T = \mathit{Ltri}(b_T)$), is denoted by q_T. See

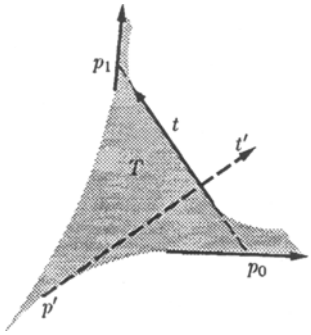

Fig. 15. Forward and backward T-views p_0 and p_1 of t cannot both have smaller slope than t.

also Fig. 18. To avoid confusion, we stress that the forward (resp. backward) view from a point along a directed line is defined with respect to the set of *obstacles* (and not with respect to the set of pseudotriangles); see Section 2.2, where this view is defined as an *obstacle*.

For later use we isolate a simple, but crucial, feature of pseudotriangles of greedy pseudotriangulations.

Lemma 16. *Let T be a pseudotriangle of a greedy pseudotriangulation.*

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 prove that no bitangent $t \in B_T$ has forward and backward T-views of smaller slope. This will prove part (1), since all points on the segment $z_T p_T$ have both forward and backward T-view of smaller slope. A similar argument proves part (2).

To prove the claim, suppose that both the backward and forward T-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 Fig. 15. Then $T = Ltri(t)$, 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 other bitangent (different from t) between $T = Ltri(t)$ and $Rtri(t)$. We give t' the direction that is compatible with the slope of its head and tail. The bitangent t' intersects 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. However, t and t' are crossing; and consequently $t' \prec t$ in contradiction with the greediness of the pseudotriangulation (claim (2) of Theorem 12). This proves the lemma. □

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 $R = Rtri(b)$ and $L = Ltri(b)$. Let $b^* = \varphi(b)$ be the bitangent obtained by flipping b. Its tail and head are denoted by p^* and q^*, respectively.

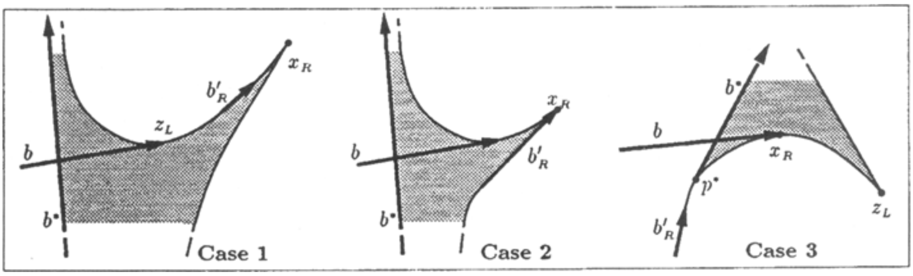

Fig. 16. The pseudotriangle $R' = Rtri(b^*)$ (shaded) is obtained by flipping bitangent b. Furthermore, R' is the left or right pseudotriangle of b'_R (cases 1 and 2, respectively), or $\partial R'$ does not contain b'_R (case 3).

The right and left pseudotriangles of b^* (with respect to the filter $I\backslash\{b\}$) are denoted by R' and L', respectively. We denote by G and G' the pseudotriangulations $G(I)$ and $G(I\backslash\{b\})$, respectively. We consider the bitangent b_T for $T = R', L'$. We only consider the pseudotriangle R' (the story for L' is completely similar). 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^*, namely the one with smaller slope. So $b^* = \min B_{R'}$, if p^* lies between b and b'_R, and $b'_R = \min B_{R'}$, otherwise. Hence there are three basic cases, that return throughout this section. They are illustrated in Fig. 16.

Case 1: *b and b'_R are not separated by a cusp of R.* Then $R' = Rtri(b'_R)$, and p^* does not 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, and p^* does not lie on the arc between b and b'_R.* Then $R' = Ltri(b'_R)$ and $\min B_{R'} = b'_R$. (Note: in this case $x_R = Head(b'_R)$, as in Fig. 16, or $x_R = Head(b)$.)

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

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 $z_{R'}$.

Case 1: *$R' = Rtri(b'_R)$.* 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 Fig. 17(a). 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^*, as illustrated in Fig. 17(b).

Case 2: *$R' = Ltri(b'_R)$ and $b'_R = \min B_{R'}$.* In this case the basepoint of R' is $Head(b'_R)$, 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; again see Fig. 17(a). 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 Fig. 17(b). Finally, the third cusp $z_{R'}$ is equal to z_L, if $Head(b) = x_R$, otherwise it is equal to x_R, as illustrated in Fig. 17(c).

Case 3: *$R' = Rtri(b^*)$ and $b^* = \min B_{R'}$.* In this case $Head(b) = x_R$, 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

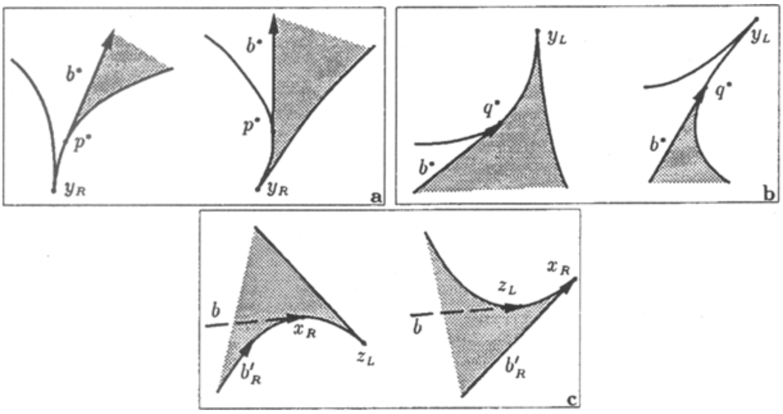

Fig. 17. The cusps of R'.

is also equal to the third cusp $z_{R'}$; see the left part of Fig. 17(a) for this situation. Since in this case x_R is a cusp of R, the second cusp $y_{R'}$ is equal to z_L, as depicted in the left part of Fig. 17(c). Finally Fig. 17(b) shows that 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.

Table 2 summarizes the previous discussion.

3.4.2. *The Splittable Queue* Awake[T]. Conceptually the flipping can be done by walking—in the positive direction, starting at the basepoint—along the boundaries of the pseudotriangles L (left) and R (right) incident upon the flipped bitangent b, with one leg in every pseudotriangle, 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 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 basepoint *Tail*(b) of R. Similarly, q^* lies between x_L and q_L.

Definition 17. For a pseudotriangle T, a point in ∂T is called *awake* if it lies between x_T and q_T.

Table 2. The cusps of R'.

	$x_{R'}$	$y_{R'}$	$z_{R'}$
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^*

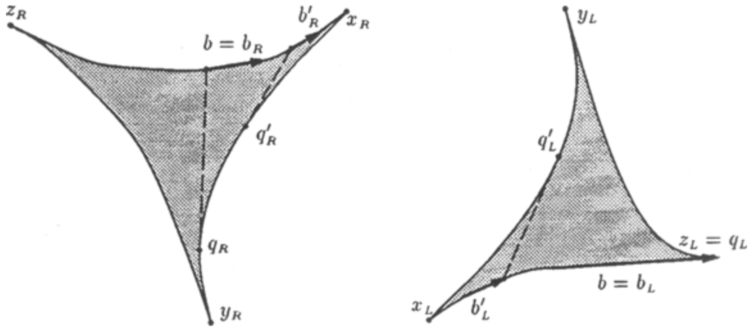

Fig. 18. The set of points that are *awake* in T is the segment $x_T q_T$, for $T = L, R$. When the algorithm flips $b = b_R = b_L$, the walk on ∂T starts in q'_T (case 1) or in the cusp x_T (cases 2 and 3).

Note that the points of ∂R that are awake have forward R-view of smaller slope, whereas the points awake in ∂L have backward L-view of smaller slope; see Fig. 18. Lemma 16 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), as shown in Fig. 18.

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 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 is 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 preceded 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 terminal points of an atom. See Section 3.4.3 for more details on this split operation.

Lemma 18. *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 see Appendix B. For our current purposes we stress that we maintain, for each pseudotriangle T, a splittable queue `Awake[T]`, satisfying the following invariant:

Invariant 1. `Awake[T]` represents the segment $x_T q_T$ of ∂T (the atom containing x_T being the head of the queue).

We now describe in more detail (i) how to compute b^*, using `Awake[R]` and `Awake[L]`, and (ii) how to restore Invariant 1 for the new pseudotriangles R' and 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, namely 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 *dequeuing* 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 `AwakeMax[L]` (namely the atoms containing p^* and q^*, respectively) by replacing their terminal points 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 use these queues in the construction of the queues `Awake[R′]` and `Awake[L′]`. We summarize the preceding discussion in the following piece of pseudocode.

Algorithm COMPUTING b^*

```
1 if $x_R$ does not separate $b_R$ and $b'_R$ then
2     Comment: case 1
3     search for arc in Awake[R], containing $q'_R$
4     split Awake[R] at $q'_R$, into AwakeMin[R] and AwakeMax[R]
```

```
5  else Comment: cases 2 and 3
6      AwakeMin[R] ← $\emptyset$
7      AwakeMax[R] ← Awake[R]
8  endif Comment: AwakeMax[R] represents $q'_R q_R$ (case 1) or $x_R q_R$ (case 2, 3)
9  Construct AwakeMin[L] and AwakeMax[L] similarly
10 Find $p^*$ and $q^*$ by synchronous linear search on AwakeMax[R] and
       AwakeMax[L], meanwhile dequeuing atoms not containing $p^*$ and $q^*$,
       respectively
11 Set initial point of first atom in AwakeMax[R] to $p^*$
12     Comment: AwakeMax[R] represents subsegment $p^* q_R$ of $\partial R$
13 Set initial point of first atom in AwakeMax[L] to $q^*$
14     Comment: AwakeMax[L] represents subsegment $q^* q_L$ of $\partial L$
```

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, namely 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. In other words, we maintain, for each pseudotriangle T, the following invariant:

Invariant 2. Asleep[T] represents the following segment of ∂T: $y_T z_T$, if $q_T \notin y_T z_T$, and $q_T z_T$, if $q_T \in y_T z_T$.

We only describe how to establish Invariants 1 and 2 for pseudotriangle R'; the invariants are established similarly for L'. In particular we show that 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], requires only a number of dequeue and *at most four enqueue* operations. Again we consider each of the cases, introduced in Section 3.4.1, separately.

Case 1: $R' = \mathit{Rtri}(b'_R)$. Since in this case $\mathit{Head}(b)$ is not a cusp of R, it is a cusp of L. Figure 16, case 1, illustrates this observation. More precisely, $\mathit{Head}(b) = z_L$. Moreover, the point of ∂L whose tangent contains the basepoint $\mathit{Head}(b)$ of ∂L, coincides with $\mathit{Head}(b)$, 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 $\mathit{Tail}(b'_R)$, 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. This justifies line 2 in the following piece of pseudocode:

Algorithm CONSTRUCTION OF Awake[R′] AND Asleep[R′]: CASE 1

```
1 Comment: Awake[L] $= x_L z_L$, Asleep[L] $= \emptyset$
2 Awake[R′] ← AwakeMin[R]
3     Comment: Invariant 1 holds for $R'$
4 if $z_{R'} = y_L$ then
5     Asleep[R′] ← AwakeMax[L]
```

```
 6     dequeue last atom from Asleep[R′]
 7  else Comment: $z_{R'} = q^*$
 8     Asleep[R′] ← ∅
 9  endif Comment: Asleep[R′] represents subsegment of $q^* z_{R'} \subset \partial R'$ of points
       asleep in $\partial R'$
10  enqueue segment $b^* = p^* q^*$ at the head of Asleep[R′]
11     Comment: Asleep[R′] represents $p^* z_{R'} \subset \partial R'$
12  if $y_{R'} = y_R$ then
13     if $q_{R'} \in x_R y_R$ then
14        enqueue arc $y_R p^*$ at the head of Asleep[R′]
15     else Comment: $q_{R'} \in y_R z_R$
16        enqueue arc $q_{R'} p^*$ at the head of Asleep[R′]
17     endif
18  else Comment: $y_{R'} = p^*$
19     skip (do nothing)
20  endif Comment: Invariant 2 holds for $R'$
```

To see how Asleep[R′] is constructed in lines 3–20, 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 ($= q_{R'}$).

Lines 4–9 initialize Asleep[R′], so that it represents the chain of points on $q^* z_{R'}$, that are asleep in R'. To see this, recall from the end of Section 3.4.3 that AwakeMax[L] represents the segment $q^* z_L$ of ∂L, since $q_L = z_L$. Furthermore, Lemma 16, part (2), tells us that the segment $y_L z_L$ is a single arc. Therefore this arc is the last atom in AwakeMax[L]. So if $z_{R'} = y_L$ (see Table 2), we initialize Asleep[R′] in line 5, after which we dequeue the last atom from this queue in line 6. If $z_{R'} = q^*$ the segment $q^* z_{R'}$ is empty, justifying the assignment in line 8.

To complete the construction, observe that b^* is asleep in R'. Therefore we enqueue, in line 10, 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′]. This case is handled in lines 18 and 19. So according to Table 2, it remains to consider the case $y_{R'} = y_R$. This is done in lines 12–17. According to Lemma 16, the segment $y_R p^* \subset y_R q_R$, is a single arc. If $q_{R'}$ ($= q'_R$) lies between x_R and y_R, all points on the arc $y_R p^*$ are asleep in R', so the first atom of Asleep[R′] should represent this arc. Finally, if $q'_R \in y_R z_R$, the first atom of Asleep[R′] should represent the arc $q'_R p^*$. In either case we enqueue an atom at the head of Awake[R′], which represents an arc with terminal point p^*, and initial point $y_{R'}$. This completes the construction of Asleep[R′] in case 1.

Case 2: $R' = Ltri(b'_R)$ *and* $b'_R = \min B_{R'}$. We distinguish two subcases.

Case 2.1: *Head*$(b) = x_R$. In this situation $z_{R'} = z_L$. To determine the part $x_{R'} q_{R'}$ of $\partial R'$ that is awake, we consider two further subcases.

Case 2.1.1: $q_{R'}$ *comes before* p^* *on* $\partial R'$. We can determine in $O(1)$ time whether this case arises by comparing the position of $p_{R'}$ with respect to the tangent at p^*. In this case, by Definition 17, b^* is not awake in $\partial R'$. Note also that in this case $x_{R'} = y_R$. Now Lemma 16, part (2), tells us that the points $x_{R'}$ ($= y_R$), $q_{R'}$, p^*, and q_R lie on a single arc (in ∂R). Consequently, the points that are awake in $\partial R'$ form a single arc $x_{R'} q_{R'}$ ($= y_R q_{R'}$).

Therefore we restore Invariant 1 for pseudotriangle R' by initializing `Awake[R′]` as the empty queue, after which we enqueue a single atom, representing the arc $x_{R'}q_{R'}$ on $\partial R'$. See lines 2–4 of the pseudocode below.

Case 2.1.2: *$q_{R'}$ comes after p^* on $\partial R'$*. In this case $q_{R'}$ lies on the segment q^*q_L of ∂L. As explained in Section 3.4.3 this segment is represented by `AwakeMax[L]`. 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 first setting `Awake[R′]` $\leftarrow$ `AwakeMax[L]`, and subsequently *dequeuing* atoms from the tail of `Awake[R′]`; see lines 6–10 of the pseudocode below. When $q_{R'}$ is found, the queue `Awake[R′]` represents the segment $q^*q_{R'}$. The construction of `Awake[R′]` is completed by enqueuing an atom representing b^* at the head, followed by enqueuing an atom representing the arc $x_{R'}p^*$ at the head in case $x_{R'} \neq p^*$, see lines 14–19 of the pseudocode. (The fact that, in the latter case, $x_{R'}p^*$ is a single arc follows from Lemma 16, part (2), applied to R'.)

Case 2.2: *Head*$(b) \neq x_R$. In this case *Head*$(b'_R) = x_R$. Furthermore $p_{R'} = q_{R'} = x_R$, so:

- The part $y_{R'}z_{R'}$ ($= y_{R'}q_{R'}$) of $\partial R'$ is a single arc; see again Lemma 16, part (2), applied to R'.
- All points on $\partial R'$ between $x_{R'}$ ($= y_R$ or p^*) and $z_{R'}$ ($= x_R$) are awake in $\partial R'$. Consequently, no point is asleep in $\partial R'$.

Since in this case $q_L = z_L$ (= *Head*(b)), we see that `AwakeMax[L]` represents the part of $\partial R'$ between q^* and z_L; see line 14 in the algorithm of Section 3.4.3. So after setting `Awake[R′]` to `AwakeMax[L]`, and adjusting the endpoint of the last atom in this queue from q_L $(= z_L)$ to x_R $(= z_{R'} = q_{R'})$, we establish that `Awake[R′]` represents the part of $\partial R'$ between q^* and $z_{R'}$. See lines 6 and 11–13 of the pseudocode below.

As in case 2.1.2, we now enqueue b^* at the head of `Awake[R′]`. In case $x_{R'} = p^*$, this completes the construction of `Awake[R′]`. In case $x_{R'} \neq p^*$ we complete the restoration of Invariant 1 by enqueuing the single arc $x_{R'}p^*$ $(= y_Rp^*)$.

We summarize the preceding discussion in the following piece of pseudocode:

Algorithm CONSTRUCTION OF `Awake[R′]`: CASE 2

```
 1 if Head$(b) = x_R$ and $q_{R'}$ comes before $p^*$ on $\partial R'$ then Comment: Case 2.1.1
 2    Awake[R′] ← $\emptyset$
 3    enqueue atom representing arc $x_{R'}q_{R'}$ onto Awake[R′]
 4    Comment: Invariant 1 holds for $R'$
 5 else Comment: case 2.1.2 or case 2.2
 6    Awake[R′] ← AwakeMax[L]
 7    if Head$(b) = x_R$ then Comment: case 2.1.2
 8       while tail atom of Awake[R′] does not contain $q_{R'}$ do
 9          dequeue tail atom of Awake[R′] endwhile
10       set terminal point of tail atom in Awake[R′] to $q_{R'}$
11    else Comment: case 2.2
12       set terminal point of tail atom in Awake[R′] to $x_R$ $(= q_{R'})$
13    endif Comment: Awake[R′] represents $q^*q_{R'}$
```

```
    enqueue $b^*$ at the head of Awake[R']
    if $x_{R'} \neq p^*$ then Comment: $x_{R'} = y_R$; see Table 2
        enqueue arc $x_{R'}p^*$ at head of Awake[R']
    else Comment: $x_{R'} = p^*$
        skip (do nothing)
    endif Comment: Invariant 1 holds for $R'$
endif
```

It remains to describe the construction of Asleep[R′], namely the sequence of points of $\partial R'$ between $y_{R'}$ and $z_{R'}$ that are not awake.

As we have observed above, in case 2.2 none of the points of $\partial R'$ is asleep, so we establish Invariant 2 for R' by setting Asleep[R′] to $\emptyset$. So consider case 2.1, namely $Head(b) = x_R$. Then $z_{R'} = z_L$, and all points that are asleep in $\partial R'$ belong to ∂L. If q_L does not belong to the part of ∂L between y_L and z_L, then $y_{R'}z_{R'} = y_L z_L$, and $q_{R'} \notin y_L z_L$, so we restore Invariant 2 for R' by setting Asleep[R′] $\leftarrow$ Asleep[L]. If, on the other hand, $q_L \in y_L z_L$, then q_L is—by definition—the initial point of the first atom of Asleep[L]. In this case Lemma 16, part (2), applied to the pseudotriangle L, tells us that $y_L q_L$ is a single arc, so we can detect in $O(1)$ time whether $q_{R'}$ lies between y_L and z_L (since, in that case, it lies on the arc $y_L q_L$ of ∂L). If $q_{R'} \in y_L z_L$, then the first atom of Asleep[R′] is the union of the arc $q_{R'}q_L$ and the arc that is the first atom of the queue Asleep[L]. In other words, after setting Asleep[R′] $\leftarrow$ Asleep[L], we can establish Invariant 2 for R' in this case by replacing the initial point (namely q_L) of the first atom in Asleep[R′] with $q_{R'}$. See line 5 of the pseudocode below. If $q_{R'} \notin y_L z_L$, we replace the initial point of the first atom of Asleep[R′] with $y_{R'}$, which is either q^* or y_L. See lines 6–8 in the pseudocode.

The preceding discussion is summarized in the following code fragment.

Algorithm CONSTRUCTION OF Asleep[R′]: CASE 2

```
 1 if $Head(b) = x_R$ then Comment: case 2.1, $z_{R'} = z_L$
 2     Asleep[R'] <- Asleep[L]
 3     if $q_L \in y_L z_L$ then
 4         if $q_{R'} \in y_L z_L$ then
 5             set initial point of first atom in Asleep[R'] to $q_{R'}$
 6         elseif $q^* \in y_L z_L$ then
 7             set initial point of first atom in Asleep[R'] to $q^*$
 8         else set initial point of first atom in Asleep[R'] to $y_L$
 9         endif
10     else skip (do nothing)
11     endif
12 else Comment: case 2.2, $z_{R'} = x_R = q_{R'}$
13     Asleep[R'] <- $\emptyset$
14 endif Comment: Invariant 2 holds for $R'$
```

Case 3: $R' = Rtri(b^*)$ *and* $b^* = \min B_{R'}$. In this case R' is the *right* pseudotriangle of b^*, so p^* is the basepoint of R', and $q_{R'} = p^*$. Again we refer to Fig. 16, 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 2.1. 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 initial point of the first arc from q_L into $x_{R'}$, which is either q^* (if $q^* \in y_L z_L$) or y_L (if $q^* \notin y_L z_L$), according to Table 2. In both cases we finally enqueue the arc $z_L p^* = y_{R'} q_{R'}$ at the tail of Awake[R'].

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 enqueuing 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], $T \in G(I(0))$, cost $O(k)$ further enqueue, dequeue, and split operations. Note that, according to Lemma 3, at any time the storage needed for all these queues is $O(n)$. Together with Lemma 18 this observation implies our main result, namely Theorem 1.

Although we have not checked all the details yet, we are convinced that this algorithm also applies to the case of obstacles of nonconstant complexity. More precisely, consider n convex objects consisting of a total of m pieces of constant complexity. Then the algorithm computes the visibility graph of this collection in time $O(k + m \log n)$ and space $O(m)$. In this case the number of enqueue operations in a single flip has an upper bound proportional to the complexity of the face of the visibility complex whose minimal vertex corresponds to the bitangent being flipped. Then the amortized complexity analysis of the algorithm can be done conveniently in terms of the combinatorial complexity, namely $O(k + m)$, of the visibility complex.

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 of constant complexity in the plane. Our algorithm realizes a topological sweep of the visibility complex and is based on new combinatorial properties of visibility graphs/complexes. As indicated in Section 3.4.5, we are convinced that the algorithm also works for obstacles of nonconstant complexity.

This work raises two questions that we intend to study in the future. The first question is whether our method can be extented to nonconvex 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 nonconvex 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 [17]).

Acknowledgments

We are greatly indebted to the anonymous referees for providing us with many helpful suggestions for improving the readability of the paper.

Appendix A. Poset Terminology

In this section we review poset terminology, that we borrow from the book of Stanley [28, Chapter 3] and the paper of McMullen [19].

A *partially ordered set* P (or *poset*, for short) is a set, together with a partial order relation denoted by $\preceq$. A *subposet* of P is a subset of P with the induced order. A special type of subposet is the *interval* $[x, y] = \{z \in P \mid x \preceq z \preceq y\}$. 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 $x \prec y$ (i.e., $x \preceq y$ and $x \neq y$) and if no element $z \in P$ satisfies $x \prec z \prec y$. 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 $z \in P\backslash C$ such that $x \prec z \prec y$ 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 I of P such that if $x \in I$ and $x \preceq y$, then $y \in I$. A *proper* filter of a poset $(X, \preceq)$ is a filter which is a nonempty proper subset of X.

An abstract n-polytope is a poset $(P, \preceq)$, 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 $F \prec G$, then there are exactly two i-faces H such that $F \prec H \prec G$. (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.

Appendix B. Splittable Queues

Here we sketch a proof of Lemma 18. It is well known that finger trees suit our purpose (see, e.g., [13]), 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. Note in passing that the randomized search trees of Aragon and Seidel [3] can also be used to implement splittable queues, resulting in the same time bounds with high probability.

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 1 (1, 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., Chapter III of [20] or Chapter 3.3 of [21]). 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. It is similarly shown that dequeuing 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, toward 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 1 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 [5] 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.

References

1. J. Abello and K. Kumar. Visibility graphs and oriented matroids. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, pages 147–158. Lecture Notes in Computer Science, volume 894. Springer-Verlag, Berlin, 1995.
2. P. K. Agarwal, N. Alon, B. Aronov, and S. Suri. Can visibility graphs be represented compactly? *Discrete Comput. Geom.*, 12:347–365, 1994.
3. C. Aragon and R. Seidel. Randomized search trees. *Proc. 30th Ann. IEEE Symp. on Foundations of Computer Science*, pages 540–545, 1989.

4. T. Asano, T. Asano, L. Guibas, J. Hershberger, and H. Imai. Visibility of disjoint polygons. *Algorithmica*, 1:49–63, 1986.
5. B. Chazelle and H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *J. Assoc. Comput. Mach.*, 39:1–54, 1992.
6. C. Coullard and A. Lubiw. Distance visibility graphs. *Internat. J. Comput. Geom. Appl.*, 2:349–362, 1992.
7. F. Durand and C. Puech. The visibility complex made visibly simple. *Video Proc. 11th Ann. ACM Symp. on Computational Geometry*, page V2, Vancouver, 1995.
8. H. Edelsbrunner and L. J. Guibas. Topologically sweeping an arrangement. *J. Comput. System Sci.*, 38:165–194, 1989. Corrigendum in 42:249–251, 1991.
9. H. N. Gabow and R. E. Tarjan. A linear-time algorithm for a special case of disjoint set union. *J. Comput. System Sci.*, 30:209–221, 1985.
10. S. K. Ghosh and D. M. Mount. An output-sensitive algorithm for computing visibility graphs. *SIAM J. Comput.*, 20:888–910, 1991.
11. 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.
12. J. Hershberger. An optimal visibility graph algorithm for triangulated simple polygons. *Algorithmica*, 4:141–155, 1989.
13. K. Hoffmann, K. Mehlhorn, P. Rosenstiehl, and R. E. Tarjan. Sorting Jordan sequences in linear time using level-linked search trees. *Inform. and Control*, 68:170–184, 1986.
14. S. Kapoor and S. N. Maheshwari. Efficient algorithms for euclidean shortest paths and visibility problems with polygonal obstacles. *Proc. 4th Ann. ACM Symp. Computational Geometry*, pages 178–182, 1988.
15. D. E. Knuth. *The Art of Computer Programming: Fundamental Algorithms*, second edition. World Student Series Edition. Addison-Wesley, Reading, MA, 1973.
16. D. E. Knuth. ⟨⟨*Literate Programming*⟩⟩. Center for Study of Language and Information, Lectures Notes, volume 27, 1992.
17. B. Korte, L. Lovász, and R. Schrader. *Greedoids*. Algorithms and Combinatorics, Number 4. Springer-Verlag, Berlin, 1991.
18. T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM*, 22(10):560–570, 1979.
19. P. McMullen. Modern developments in regular polytopes. In T. Bisztriczky, P. McMullen, R. Schneider, and A. Ivić Weiss, editors, *Polytopes: Abstract, Convex and Computational*, pages 97–124. NATO ASI Series, volume 440. Kluwer, Dordrecht, 1994.
20. K. Mehlhorn. *Data Structures and Algorithms 1: Sorting and Searching*. EATCS Monographs on Theoretical Computer Science, volume 1. Springer-Verlag, Berlin, 1984.
21. B. M. E. Moret and H. D. Shapiro. *Algorithms from P to NP*, volume I. Benjamin/Cummings, Redwood City, CA, 1990.
22. J. O'Rourke. Computational geometry column 18. *Internat. J. Comput. Geom. Appl.*, 3(1):107–113, 1993.
23. M. H. Overmars and E. Welzl. New methods for computing visibility graphs. *Proc. 4th Ann. ACM Symp. on Computational Geometry*, pages 164–171, 1988.
24. M. Pocchiola and G. Vegter. The visibility complex. *Proc. 9th Ann. ACM Symp. on Computational Geometry*, pages 328–337, 1993. Full version to appear in *Internat. J. Comput. Geom. Appl.*, 1996.
25. M. Pocchiola and G. Vegter. Order types and visibility types of configurations of disjoint convex plane sets (extented abstract). Technical Report 94-4, Labo. d'Inf. de l'ENS, Paris, January 1994.
26. M. Pocchiola and G. Vegter. Minimal tangent visibility graphs. *Comput. Geom. Theory Appl.*, 1996. To appear.
27. H. Rohnert. Time and space efficient algorithms for shortest paths between convex polygons. *Inform. Process. Lett.*, 27:175–179, 1988.
28. R. P. Stanley. *Enumerative Combinatorics*, volume 1. Wadsworth & Brooks/Cole, Monterey, CA, 1986.
29. S. Sudarshan and C. P. Rangan. A fast algorithm for computing sparse visibility graphs. *Algorithmica*, 5:201–214, 1990.
30. E. Welzl. Constructing the visibility graph of n line segments in the plane. *Inform. Process. Lett.*, 20:167–171, 1985.

Received April 28, 1995, *and in revised form December* 11, 1995, *and April* 9, 1996.