Computing the Visibility Graph via Pseudo–triangulations

Michel Pocchiola* Gert Vegter†

Abstract

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

1 Introduction

Consider a collection $\mathcal{O}$ of pairwise disjoint convex objects in the plane. We are interested in problems in which these objects arise as obstacles, either in connection with visibility problems where they can block the view from an other geometric object, or in motion planning, where these objects may prevent a moving object from moving along a straight line path. The *visibility graph* is a central object in such contexts. For polygonal obstacles the vertices of these polygons are the nodes of the visibility graph, and two nodes are connected by an arc if the corresponding vertices can see each other. [9] describes the first non-trivial algorithm for computing the visibility graph of a polygonal scene with a total of n vertices in $O(n^2)$ time. [4] presents an optimal $O(n \log n + k)$ algorithm, where k is the number of arcs of the visibility graph. A practically feasible $O(k \log n)$ algorithm is contained in [6].

In this paper we present an optimal—with respect to both time and working space—algorithm that computes the *tangent visibility graph* of $\mathcal{O}$. 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). The endpoints of these bitangents split the boundaries of the obstacles into a sequence of arcs; these arcs and the free bitangents are the edges of the tangent visibility graph, see Figure 1.

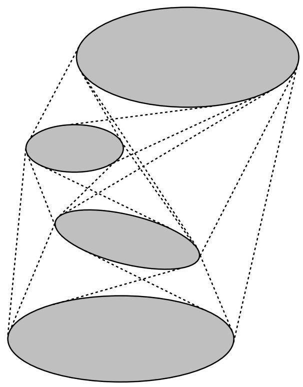

Figure 1: The tangent visibility graph.

In [7] we described an optimal method for computing the so–called visibility complex of the collection $\mathcal{O}$. Just as the algorithm of Ghosh and Mount, see [4], it is based on complicated data structures (e.g. the split–find structure of Gabow and Tarjan, see [3]). Therefore it is not suitable for a practical implementation.

We give two practical, yet efficient methods to compute the tangent vibility graph of a collection of n disjoint convex sets in the plane in output sensitive time. The first algorithm is very simple, and uses

*e-mail: pocchiol@dmi.ens.fr
†e-mail: gert@cs.rug.nl

0

$O(k \log n)$ time, where k is the number of arcs of the tangent visibility graph. Throughout the paper we assume that the complexity of the objects is $O(1)$, that is, the common bitangents of any pair of objects can be computed in constant time. With each unit vector in the plane we associate a subdivision of free space, which we call the *greedy pseudo–triangulation* associated with this vector. The algorithm maintains the greedy pseudo–triangulation as the unit vector rotates over an angle of π. The basic operation that updates the pseudo–triangulation is a *flip* of a free bitangent with smallest slope greater than the slope of the rotating unit vector. Relaxing the order in which bitangents are flipped we obtain an optimal algorithm, using $O(k + n \log n)$ time and $O(n)$ working space. (To the best of our knowledge, even for the case of line segments, this is the first optimal algorithm that uses linear working space.)

If the obstacles are points, our second method—translated into dual space—is an alternative for the topological sweep algorithm for arrangements of lines, of Edelsbrunner and Guibas, see [2]. Our pseudotriangulations replace their (upper and lower) horizon trees.

It turns out that, in general, our second method can be interpreted as a topological sweep of the visibility complex, introduced in [7]. This point is briefly discussed in the last section.

2 Greedy pseudo–triangulations

Definition and basic properties

Let $\mathcal{O} = \{O_1, O_2, \ldots, O_n\}$ be a family of n pairwise disjoints convex sets (obstacles for short). A *pseudo–triangulation* of a set of obstacles is the subdivision of the plane induced by a maximal (with respect to inclusion) family of pairwise noncrossing free bitangents. It is clear that a pseudo–triangulation always exists and that the bitangents of the boundary of the convex hull of the obstacles are edges of any pseudo–triangulation. A pseudo–triangulation of a collection of four obstacles is depicted in Figure 2. The subdivision owes its name to the special shape of its regions. A *pseudotriangle* is a simply connected subset T of the plane, such that (i) the boundary ∂T consists of three convex chains, that share a tangent at their common endpoint, and (ii) T is contained in the triangle formed by the three endpoints of these convex chains. These three endpoints will be called the *cusps* of T. (In this paper a chain is an alternating sequence of free bitangents and arcs, such that successive elements share a common endpoint, at which the bitangent is tangent to the arc.) Without proof we mention the following result (it is easy to prove using Euler's relation for planar graphs, see the full version).

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

Consider a unit vector u in the plane. The u–slope of a directed line segment b is defined as the *positive* (counterclockwise) angle over which we have to rotate u in order to obtain a vector parallel to b. The *greedy* pseudo–triangulation, associated with a unit vector u, is the pseudo–triangulation induced by the family $B(u) = \{b_1, b_2, \ldots, b_{3n-3}\}$ of bitangents, recursively defined as follows.
1. b_1 is the bitangent with smallest u–slope in the set of free bitangents.
2. b_{i+1} is the bitangent with smallest u–slope in the set of free bitangents disjoint from $b_1, b_2, \ldots, b_i$.
Figure 2 depicts a greedy pseudo-triangulation. The greedy pseudo–triangulation associated with u, will be denoted by $\mathcal{T}(u)$.

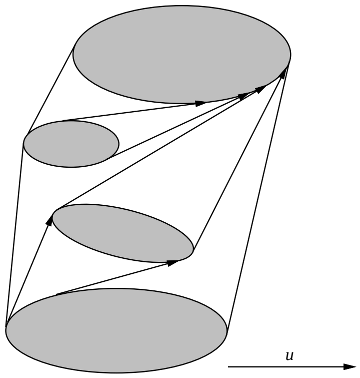

Figure 2: The greedy pseudo-triangulation $\mathcal{T}(u)$ (with respect to u).

If t is a free bitangent, then either $t \in B(u)$, or t intersects at least one bitangent of $B(u)$ whose u–slope is less than the u–slope of t. This property holds for *all* bitangents in $B(u)$, intersecting t:

Proposition 2 *The u–slope of a free bitangent t, $t \notin B(u)$, is larger than the u–slope of every bitangent in the sequence $B(u)$, intersecting t.*

Proof. Suppose the result does not hold. Let t be a free bitangent of minimal u–slope, intersecting a bitangent in $B(u)$ of larger u–slope. As we have just observed, there also is a bitangent in $B(u)$, intersecting t, of smaller u–slope than t. In particular, there are $b, b' \in B(u)$ intersecting t, such that (i) the u–slope of t is greater than the u–slope of b, but less than the u–slope of b', and (ii) there is no bitangent in $B(u)$ intersecting t between its points of intersection with b and b'. In other words: b and b' are in the boundary of the same pseudo–triangle in $\mathcal{T}(u)$. Let us denote this pseudotriangle by T.

Consider the point $q \in \partial T$ whose tangent line is parallel to t. If t intersects b before (after) b', the point q lies to the left (right) of t, see Figure 3. Let p be the tail (head) of b.

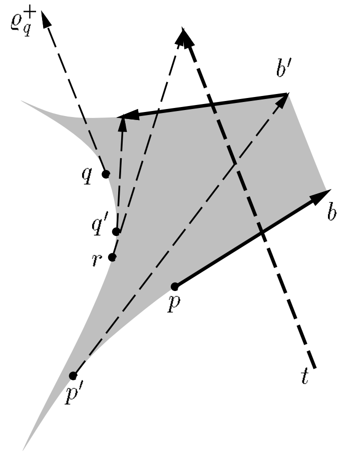

Figure 3: Proof of the basic property, in case t intersects b before b'.

For a point x in the boundary of T let ϱ_x^+ (ϱ_x^-) be the directed free line segment starting (terminating) at x, and extending in forward (backward) direction along the tangent line of ∂T at x, until it hits some obstacle. This object is called the visibility of x along the ray.

As x moves from p to q along ∂T, let p' be the first and q' be the last point on ∂T for which the corresponding ray intersects the bitangent b'. Note that p and p', as well as q and q', may coincide. Furthermore ϱ_x^+ (ϱ_x^-) intersects b' for all points $x \in \partial T$ between p' and q'. As x travels along ∂T from p' to q', the u–slope of ϱ_x^+ is increasing, so in particular it is less than the u–slope of t. We shall argue that, as a point x moves along ∂T from p' to q', the object visible from x along ϱ_x^+ (ϱ_x^-) changes. Suppose we know this is true, then there is a point $x \in \partial T$ between p' and r, such that ϱ_x^+ contains a point of tangency y. Therefore xy is a free bitangent intersecting b', whose u–slope is larger than the u–slope of b'. Since t is the free bitangent with minimal u–slope satisfying this property, the u–slope of xy is smaller than the u–slope of t. But we just observed that the u–slope of ϱ_x^+, and hence the u–slope of xy, is smaller than the u–slope of t. This is a contradiction.

So it remains to prove that the visibility along ϱ_x^+ is not constant. We only do so in the case where t intersects b before b'. The other case is treated similarly. Assume that we see the same object, O' say, along $\varrho_{p'}^+$ and $\varrho_{q'}^+$ (otherwise we're done). Let I be the open line segment connecting the endpoints of these rays, then $I \subset O'$, so in particular I and t are disjoint.

Let $r \in \partial T$ be the point where the tangent line l_r of ∂T through r contains the endpoint of t. Obviously r lies between p' and q on ∂T. It even lies between p' and q'. Indeed, if this were not the case then $\varrho_{q'}^+$ would end at the head of b', which would therefore be a point of I. Then the line supporting $\varrho_{q'}^+$ would intersect I before t. On the other hand, the line supporting $\varrho_{p'}^+$ obviously intersects t before I. Since the line supporting ϱ_x^+ intersects both t and I, for all $x \in \partial T$ between p' and q', the segments t and I would not be disjoint. This contradiction proves that r lies between p' and q'.

Let O'' be the object containing the endpoint of t. Since I and t are disjoint, the line l_r intersects O'' before O', so the object visible from r along ϱ_r^+ is different from O'. $\square$

Lemma 3 *The greedy pseudo–triangulation of a collection of n disjoint convex obstacles in the plane with respect to some unit vector can be computed in $O(n \log n)$ time.*

Proof. Omitted from this version. 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. The $O(n)$ events correspond to the detection of free bitangents of the greedy pseudo–triangulation. $\square$

3 The greedy flip algorithm

The idea of the first version of the algorithm is very simple: just maintain $\mathcal{T}(u)$ as u rotates over an angle of π, starting from the horizontal direction u_0. It is obvious that $\mathcal{T}(u)$ remains constant as long as it is not parallel to any of the free bitangents of $\mathcal{O}$. It turns out that we can obtain all greedy pseudo–triangulations of the collection $\mathcal{O}$ by *flipping* the bitangent of minimal slope with respect to the current unit vector. To make this idea more precise, consider two pseudotriangles T_1 and T_2 that share a bitangent,

b say. We obtain a new pseudo–triangulation by *flipping* b, i.e. by replacing b by the common bitangent of T_1 and T_2. (To see that this common bitangent is unique, observe that two distinct tangent lines of ∂T_i cross inside T_i.) E.g. in Figure 4, flipping b_1 amounts to replacing it by b^* (here T_1 and T_2 are the shaded regions incident upon $b = b_1$). Flipping a bitangent in the boundary of the convex hull boils down to reverting its direction.

Lemma 4 *Let b be the bitangent of $\mathcal{T}(u)$ of minimal u–slope. Let u' be a unit vector obtained by infinitesimally rotating u beyond b. Then $\mathcal{T}(u')$ is obtained from $\mathcal{T}(u)$ by flipping the bitangent b.*

Proof. Let $B(u) = \{b_1, \cdots, b_{3n-3}\}$ and $B(u') = \{b'_1, \cdots, b'_{3n-3}\}$. Furthermore let b^* be the free bitangent obtained by flipping b_1. First assume that b_1 is an internal bitangent, i.e. it is not in the boundary of the convex hull of the collection of objects. Then there is an index i, with $1 \leq i < 3n-3$, such that $b'_j = b_{j+1}$, for $1 \leq j < i$, and $b'_i \neq b_{i+1}$. We shall successively prove:
(i) b'_i intersects b_1.
(ii) u–slope$(b'_i) \leq u$–slope(b^*).
(iii) $b'_i = b^*$.
(iv) $b'_j = b_j$, for $i < j \leq 3n-3$.
This will obviously prove the lemma. To prove (i), assume that b'_i and b_1 are disjoint. Since $\mathcal{T}(u')$ is a greedy pseudo–triangulation, that b'_i is the bitangent with smallest u'–slope in the set of free bitangents disjoint from $b_2, \ldots, b_{i-1}$, and hence also in the set of free bitangents disjoint from $b_1, \ldots, b_{i-1}$. But then $b'_i = b_i$, since $\mathcal{T}(u)$ is a greedy pseudo–triangulation. This contradiction proves (i).

Since b^* is disjoint from all bitangents in $\{b_2, \ldots, b_i\} = \{b'_1, \ldots, b'_{i-1}\}$, and b'_i is the free bitangent of smallest u'–slope among the free bitangents that are disjoint from $b'_1, \ldots, b'_{i-1}$, we see that u–slope$(b'_i) \leq u$–slope(b^*), which proves (ii).

To prove (iii), assume that $b^* \neq b'_i$. Since b'_i intersects b_1, b'_i must intersect the boundary of the quadrangle Q obtained by merging the pseudotriangles of $\mathcal{T}(u)$, incident upon b_1, see figure 4.

Note that the bitangents in ∂Q, whose u–slope is larger than the u–slope of b^*, lie either between the heads or between the tails of b_1 and b^*. The crucial observation is that b'_i intersects only bitangents in the boundary of ∂Q whose u–slope is *less than* the u–slope of b'_i, see proposition 2. In particular b'_i is disjoint from this part of ∂Q (note that it also can't be tangent to this part of ∂Q, since its u–slope does not exceed the u–slope of b^*). But then b'_i intersects b^* from right to left (note that, in view of (i), it intersects b_1 from right to left). This is a contradiction with (ii),

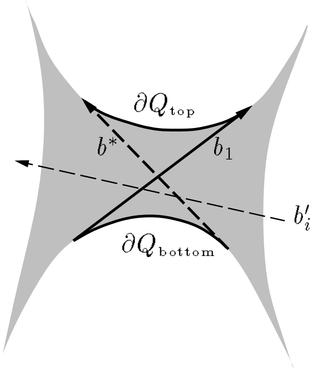

Figure 4: The pseudo–quadrangle Q, and its diagonals b_1 and b^*. Bitangent b^* is obtained by flipping b_1.

so $b^* = b'_i$.

Finally (iv) is an immediate consequence of (i), (ii) and (iii). The case in which b_1 is in the boundary of the convex hull is obvious, since flipping b_1 here amounts to reverting its direction. $\square$

Lemma 4 suggests a simple algorithm: conceptually we rotate a unit vector u, starting at position u_0, over an angle π. We maintain the set of bitangents in the current pseudo–triangulation in a priority queue, where the weight of a bitangent in the queue is its u–slope. As long as the queue is non–empty, extract the minimal weight bitangent, flip it, and insert the new bitangent into the queue if its u_0–slope is less than π (so it has not yet been detected). We shall call this method the *greedy flip algorithm*.

In this way the total time for the operations on the queue is $O(k \log n)$, since every free bitangent is deleted from the queue exactly once. The total cost of all flips is $O(k)$ (amortized). This will become clear in the next section, where we prove a more general result. Summarizing we have:

Theorem 5 *The greedy flip algorithm computes the tangent visibility graph of a collection of n disjoint convex objects in the plane in $O(k \log n)$ time, where k is the number of free bitangents.*

4 The weakly greedy flip algorithm

In this section we improve the time complexity of the algorithm by relaxing the constraint that bitangents are flipped in order of increasing slope. To achieve this goal we enlarge the class of pseudo–triangulations, and replace the linear order, induced by the slope of the bitangents, by a *partial order* on the set of bitangents in the pseudo–triangulation, such that the set of bitangents that are candidates for flipping can be maintained in constant time per flip. Furthermore, the class of pseudo–triangulations should be invariant under flipping of candidate edges. The crucial feature of this class is the property proved in proposition 2, which we now require to hold *by definition*. We then prove invariance under flipping, and describe an efficient implementation of the flip–operation, whose amortized cost is finally analyzed.

Weakly greedy pseudo–triangulations

First we need some terminology. Let B be a set of free bitangents. For a subset A in the plane the set of elements of B intersecting A is denoted by B_A. So if B is the set of all bitangents of a pseudo–triangulation $\mathcal{T}$, and T is a pseudotriangle of $\mathcal{T}$, the set B_T consists of all bitangents in ∂T. In this case the pseudotriangle of $\mathcal{T}$ incident upon $b \in B$ and —locally—to the left (right) of b is denoted by $ltriang(b)$ ($rtriang(b)$).

Consider a pseudotriangle T, and fix some (directed) bitangent $b_T \in B_T$. The direction of the tangent line in a point of ∂T is uniquely determined by the requirement that its b_T–slope is less than π. This b_T–slope is also called the *slope* of this point. The *base–point* of T, denoted by p_T, is the tail of b_T, if $T = rtriang(b_T)$, or the head of b_T, if $T = ltriang(b_T)$. (It is the unique point on ∂T at which the slope is not well–defined; by definition, we set this slope equal to 0). The positive (negative) orientation of ∂T corresponds to increasing (decreasing) slope. A subsegment of ∂T with positive (negative) orientation will be called a *walk* (*reverse walk*) along ∂T. In particular, the walk starting at the base–point of T defines a linear order on the set of bitangents B_T, called the *slope order* (with respect to b_T). The successive cusps we pass during a walk starting at the base–point of T, are denoted by x_T, y_T and z_T. The *forward* and *backward* view of point p in ∂T are the points of intersection of ∂T with the tangent line at p, lying ahead and behind p, respectively. The point whose forward (backward) view is p_T, if $T = rtriang(b_T)$ ($T = ltriang(b_T)$) is denoted by q_T.

Definition 6 *A pseudo–triangulation $\mathcal{T}$ is called* **weakly greedy** *if there is a partial order $\prec$ on its set B of bitangents such that*
(i) for every pseudotriangle T in $\mathcal{T}$ the restriction of $\prec$ to B_T is a linear order $\prec_T$, that corresponds to the slope order with respect to the minimal element of B_T.
(ii) Every free bitangent t can be given a unique direction that is compatible with the slopes of both its endpoints, such that all bitangents in B_t intersect t from left to right. This unique direction will be called the **canonical direction** *of t (with respect to $\prec$).*

Obviously proposition 2 tells us that every greedy pseudo–triangulation is weakly greedy. If T is a pseudotriangle of a weakly greedy pseudo–triangulation $\mathcal{T}$ with partial order $\prec$, we denote the minimal element of B_T with respect to $\prec$ by b_T. We say that T is a weakly greedy pseudotriangle if it is a pseudotriangle in a weakly greedy pseudo–triangulation. For later use we isolate a simple, but crucial feature of weakly greedy pseudotriangles.

Lemma 7 *Let T be a weakly greedy pseudotriangle.*
1. If $z_T \neq p_T$, then the part of ∂T between z_T and p_T is an arc.
2. If y_T lies between x_T and q_T, then the part of ∂T between y_T and q_T is an arc (i.e. it contains no bitangents).

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

To prove the claim, suppose that both the backward and forward view, p_0 and p_1 say, of t have smaller slopes than t. We only consider the case in which p_0 has smaller slope than p_1, see Figure 5. Then $T = ltriang(t)$, and the part of ∂T between p_0 and p_1 lies completely to the left of the line supporting t.

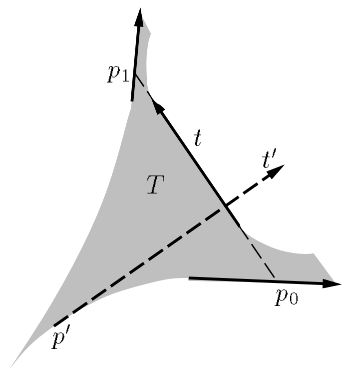

Figure 5: Backward and forward views p_0 and p_1 of t can't both have smaller slope than t.

Observe that the object containing $tail(t)$ is different from the one containing $head(t)$. Arguing as in the proof of proposition 2, we can show that there is a free bitangent t', intersecting t, whose tail p' is a point on ∂T between p_0 and p_1. But t intersects t' from right to left, in contradiction with the weakly greedyness of the pseudo–triangulation. This proves the lemma. □

Flipping $\prec$–minimal bitangents

If we work in the class of weakly greedy pseudo–triangulations we can, in general, flip more bitangents than just the one with minimal u_0–slope, without disturbing the weakly greedyness. (From now on u_0 will be a fixed, say horizontal, direction.) More precisely, we shall prove that any $\prec$–minimal bitangent can be flipped.

The partial order $\prec^*$

To introduce the partial order on the new pseudo–triangulation, consider a $\prec$–minimal bitangent b, with $R = rtriang(b)$ and $L = ltriang(b)$. Let b^* be the bitangent obtained by flipping b, and let $\mathcal{T}^*$ be the pseudo–triangulation after the flip. The right and left pseudotriangles of b^* are denoted by R' and L', respectively. The partial order $\prec^*$ on the set of bitangents of $\mathcal{T}^*$ is the transitive closure of the relation, defined by the collection of linear orders $\prec^*_T$, for $T \in \mathcal{T}^*$. If $T \neq R', L'$, then T is a pseudotriangle in both $\mathcal{T}$ and $\mathcal{T}^*$, and we take $\prec^*_T$ equal to $\prec_T$. So it remains to define $\prec^*_T$ for $T = R', L'$.

First consider the pseudotriangle R'. Let b'_R be the $\prec$–successor of b in B_R. The $\prec^*$–minimal element of $B_{R'}$ is one of the bitangents b'_R and b^*, viz the one with minimal b–slope. So $b^* = \min B_{R'}$, if $p^* = tail(b^*)$ lies between b and b'_R, and $b'_R = \min B_{R'}$, otherwise. Hence there are three basic cases, that will return throughout this section, see Figure 6.

Case 1 b and b'_R are not separated by a cusp of R.
Then $R' = rtriang(b'_R)$, and p^* doesn't lie on the arc between b and b'_R. Therefore $\min B_{R'} = b'_R$.
Case 2 b and b'_R are separated by a cusp of R, but p^* doesn't lie on the arc between b and b'_R.
Then $R' = ltriang(b'_R)$ and $\min B_{R'} = b'_R$. (Note: in this case $x_R = head(b'_R)$, as in Figure 6, or $x_R = head(b)$.)
Case 3 b and b'_R are separated by a cusp of R, but p^* lies on the arc between b and b'_R.
Then $R' = rtriang(b^*)$ and $\min B_{R'} = b^*$.

The restriction of $\prec^*$, restricted to $B_{L'}$, is defined similarly.

To make sure that the flipping terminates, we only flip bitangents whose u_0–slope is less than π. This condition, as it turns out, guarantees that the partial order, restricted to B_T, for $T \in \mathcal{T}$, is compatible with the linear order according to increasing u_0–slope.

Lemma 8 *Let $(\mathcal{T}, \prec)$ be a weakly greedy pseudo–triangulation. Let b be a $\prec$–minimal bitangent of $\mathcal{T}$, whose u_0–slope is less than π. Then the pseudo–triangulation $(\mathcal{T}^*, \prec^*)$, obtained by flipping b, is again weakly greedy. Furthermore, the canonical direction of all free bitangents t, $t \neq b$, doesn't change due to the flip, whereas the canonical direction of b is reversed.*

Proof. The proof is built from ingredients of the proof of lemma 4. We refer to figure 4 for an illustration of the proof, with the understanding that b is identified with b_1.

Since the u_0–slope of b is less than π, the relation $\prec^*$ is compatible with the order according to increasing u_0–slope, so its transitive closure is a partial order. Therefore it remains to prove that for all free bitangents $t \neq b, b^*$, the canonical direction of t with respect to $\prec^*$ is well–defined, and that the 'left–to–right' property holds. To this end it is sufficient to prove that for all such bitangents $t \neq b, b^*$, having exactly one endpoint on $\partial L \cup \partial R$, the slope at that endpoint doesn't change due to flipping b. Consider a bitangent t, having exactly one endpoint on ∂R. Assume, by contradiction, that the slope at this endpoint changes upon flipping b. Now all points of ∂R, whose slope is reversed after flipping b, lie on the arc c between b and its successor b'_R in B_R. Therefore t is tangent to c. In particular the slope of t is less

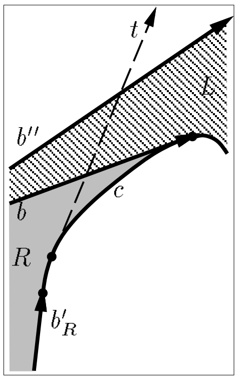

Figure 7: All free bitangents, $\neq b$, keep the same canonical direction.

than the slope of b'_R, and hence less than the slope of all bitangents in $B_R \setminus \{b\}$. By definition 6.(ii), t is therefore disjoint from all bitangents in $B_R \setminus \{b\}$, so t intersects ∂R in a point of b. Since, again by definition, b intersects t from left to right, we conclude that $tail(t) \in c$, see Figure 7. Since all points of ∂R, whose slope is reversed, lie between the base-

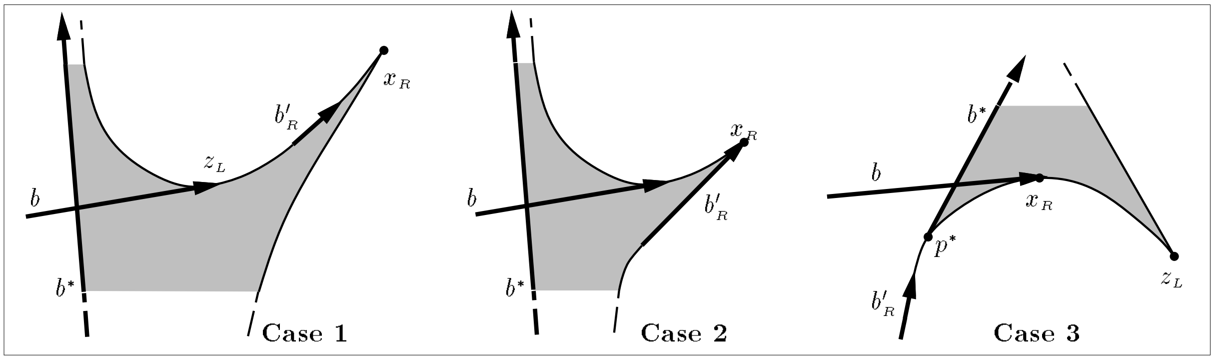

Figure 6: The partial order $\prec^*$, restricted to the bitangents of $\partial R'$. (Note that in case 2 either $x_R = head(b'_R)$, as in the Figure, or $x_R = head(b)$.)

point of R and the basepoint of $R' = rtriang(b^*)$, even $slope(t) < slope(b^*)$.

Now all bitangents t', satisfying (i) $t' \cap b \neq \emptyset$, (ii) $tail(t') \in \partial R$, and (iii) $slope(t) < slope(t')$, have their head to the left of the line supporting t, see Figure 7. Therefore $head(b^*)$ lies to the left of this line. On the other hand t is different from both b and b^*, so it intersects ∂L in a bitangent b'', different from b. Since b'' also intersects t from left to right, all points of ∂L to the left of t have slope between $slope(b)$ and $slope(b'')$. Therefore $slope(b^*) < slope(b'')$, and hence $slope(b^*) < slope(t)$. This contradiction proves that the slope at $tail(t)$ is not reversed. Similarly one can prove that the slope at $head(t)$ is not affected by the flipping of b.

We finally have to prove that either b^* and t are disjoint, or b^* intersects t from left to right. So assume $b^* \cap t \neq \emptyset$. As in the proof of lemma 4, let Q be the quadrangle obtained by merging the pseudotriangles L and R, see figure 4. Let ∂Q_{top} ($\partial Q_{\text{bottom}}$) be the part of ∂Q between the heads (tails) of b and b^*. Since $\mathcal{T}$ is weakly greedy, t is disjoint from $\partial Q_{\text{top}} \cup \partial Q_{\text{bottom}}$, since otherwise it would intersect the bitangents in this subset of ∂Q from left to right. Since t intersects b from right to left, it therefore also intersects b^* from right to left. The preceeding argument also shows that the 'left–to–right' property holds. □

The pseudotriangles R' and L'
We now consider the pseudotriangle R' in more detail, in particular its cusps $x_{R'}$, $y_{R'}$ and $z_{R'}$. (The story for L' is completely similar.) To this end we consider each of the cases 1–3 introduced above see also Figure 6.
Case 1 $R' = rtriang(b'_R)$.
In this situation b and b'_R are not separated by a cusp, so $x_{R'} = x_R$. Furthermore, if p^* lies between x_R and y_R, then the second cusp $y_{R'}$ is equal to p^*, otherwise it is equal to y_R, see Figure 8a. Similarly the third cusp $z_{R'}$ is equal to y_L, if q^* lies between x_L and y_L, otherwise it is equal to q^*, see Figure 8b.
Case 2 $R' = ltriang(b'_R)$ 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 is equal to p^*, if p^* lies between x_R and y_R, otherwise it is equal to y_R, see Figure 8a. Similarly the second cusp $z_{R'}$ is equal to y_L, if q^* lies between x_L and y_L, otherwise it is equal to q^*, see Figure 8b. Finally the third cusp $z_{R'}$ is equal to z_L, if $head(b) = x_R$, otherwise it is equal to x_R, see Figure 8c.
Case 3 $R' = rtriang(b^*)$ 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 is also equal to the third cusp $z_{R'}$, see the left part of Figure 8a. Since in this case x_R is a cusp of R, the second cusp is equal to z_L, see the left part of Figure 8c. Finally the first cusp is equal to y_L or q^*, depending on whether q^* lies between y_L and z_L or between x_L and y_L, see figure 8b. Figure 9 summarizes the previous discussion.

The algorithm

Every pseudotriangle in a weakly greedy pseudo–triangulation has a unique minimal bitangent. If a bitangent is minimal for both its left and right pseudotriangles, lemma 8 guarantees that it can be flipped. Such bitangents are called candidates.

Definition 9 *A bitangent, belonging to a weakly greedy pseudo–triangulation $(\mathcal{T}, \prec)$, is called a* **candidate** *if it is a minimal element with respect to $\prec$, and its u_0–slope is less than π.*

Lemma 8 suggests a very simple algorithm. It maintains the set of candidates in a *set* $\mathcal{C}$:

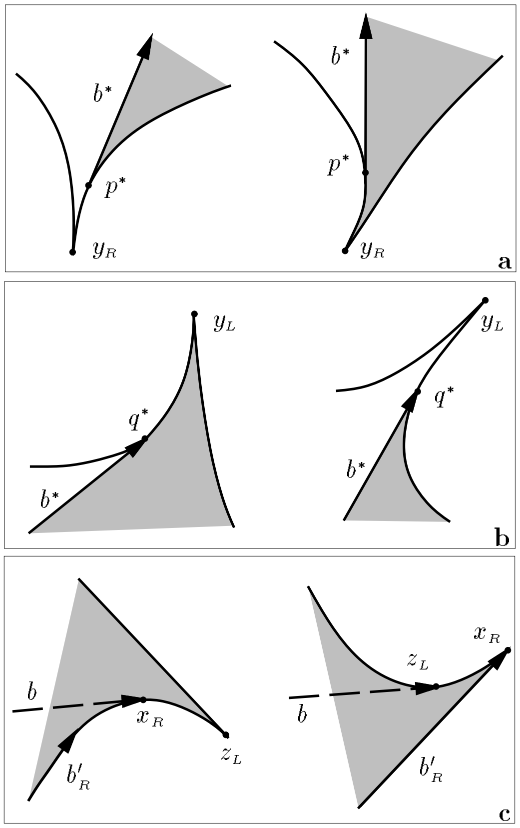

Figure 8: The cusps of R'.

1 compute the greedy pseudo–triangulation with respect to the horizontal direction u_0;
2 put all candidate edges in a *set* $\mathcal{C}$
3 **while** $\mathcal{C} \neq \emptyset$ **do**
4 select a bitangent b from $\mathcal{C}$;
5 flip b;
6 update $\mathcal{C}$;

The major improvement is that we abandoned the priority queue in favor of any simple data structure for the representation of sets, that allows us to insert and delete an element in $O(1)$ time. Of course we still have to prove that the algorithm is correct, and

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

Figure 9: The cusps of R'.

that the total time needed for flipping (viz step 5) and updating the set of candidates (viz step 6) is $O(k)$. We shall say that the algorithm *detects* a free bitangent at the moment it is flipped (in step 5). The correctness of the algorithm follows from:

Lemma 10 *The weakly greedy algorithm detects every free bitangent (or, equivalently, every edge of the tangent visibility graph).*

Proof. Let $(\mathcal{T}, \prec)$ and $(\mathcal{T}^*, \prec^*)$ be the initial and final pseudo–triangulations, respectively. For every free bitangent t the canonical direction of t with respect to $\prec$ lies between u_0 and $-u_0$. whereas its canonical direction with respect to $\prec^*$ lies between $-u_0$ and u_0. Therefore lemma 8 implies that t has been flipped. $\square$

The splittable queue `Awake[T]`

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

Observe that the tail p^* of b^* lies between the first cusp x_R and the point q_R, whose tangent contains the base–point *tail*(b) of R. Similarly q^* lies between x_L and q_L. For a pseudotriangle T, a point in ∂T is called *awake* if it lies between x_T and q_T. Note that the points of ∂R that are awake have forward view of smaller slope, whereas the points awake in L have backward view of smaller slope, see Figure 10. Lemma 7 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 (so case 1 occurs), the point p^* lies even between q'_R and q_R, where q'_R is the point whose tangent contains *tail*(b'_R), see Figure 10.

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. 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

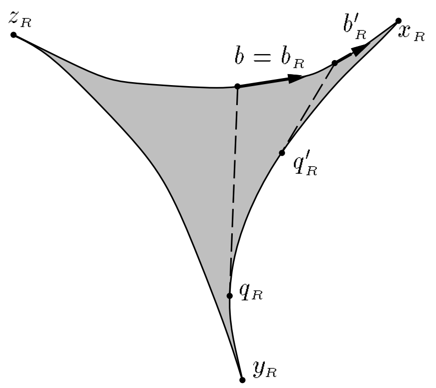

Figure 10: The set of points that are *awake* in R is the segment $x_R q_R$. When the algorithm flips $b = b_R$, the walk on ∂R starts in q'_R (case 1), or in the cusp x_R (cases 2 and 3).

atoms are in slope order. This sequence will be represented by a *splittable queue*, denoted by `Awake[T]`, a data structure for ordered lists that allows for the following operations:

(i) *enqueue* an atom, either at the head or at the tail of the list;
(ii) *dequeue* the head or the tail of the list;
(iii) *split* the sequence at an atom x; this split is preceeded by a *search* for the atom x.

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

Lemma 11 *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.

Splittable queues can be implemented using red–black trees with parent–pointers, where atoms are stored in the leaves. Splittable queues are in fact a special case of finger trees, implemented as level–linked red–black trees (see [5], and also [1] for similar ideas). In our case we don't need level links, however, since the search implicit in the split operation can be implemented as a dovetailing search up the ridges of the tree, starting from the minimal and maximal leaf of the tree. For more details and a sketch of the proof we refer to the full version of this paper.

We now describe in some detail (i) how to compute b^*, using `Awake[R]` and `Awake[L]`, and (ii) how to compute the queues `Awake[R']` and `Awake[L']`. We shall argue that doing all flips and maintaining the collection of queues `Awake[T]`, $T \in \mathcal{T}$, cost $O(k)$ enqueue, dequeue and split operations. Hence the total cost of (i) and (ii) is $O(k)$, see Lemma 11.

Construction of b^*.
If b and its successor b'_R in B_R are not separated by the cusp x_R of R, then the walk along ∂R starts in q'_R. In this case we *split* `Awake[R]` at q'_R into `AwakeMin[R]` and `AwakeMax[R]`, where the atoms in the former queue have smaller slope than the atoms in the latter queue. Otherwise, viz if b and b'_R are separated by the cusp x_R, the walk along ∂R starts in x_R, and 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]`.

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

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

Construction of `Awake[R']` and `Awake[L']`
We only describe the construction of `Awake[R']` if for both R' and L' case 1 occurs, viz when $R' = rtriang(b'_R)$ and $L' = ltriang(b'_L)$. In this situation $head(b) = z_L = q_L$, see Figure 6a, and also $tail(b) = z_R = q_R$. Furthermore, the basepoint of R'

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

If case 2 or 3 occurs for R or L, the computation of `Awake[R']` and `Awake[L']` is even simpler: It requires only a number of dequeue and *at most two* enqueue operations on the splittable queues `AwakeMax[R]` or `AwakeMax[L]`. (The full version contains further details.)

As for the amortized time complexity, observe that the initial collection of splittable queues—one for each pseudotriangle in the greedy pseudotriangulation we start out with—can be computed in $O(n \log n)$ time (for instance simply by enqueueing the bitangents and arcs, that are awake in the boundary of each pseudotriangle). This amounts to $O(n)$ enqueue–operations. As we have just indicated, doing all flips and maintaining the collection of queues `Awake[T]`, $T \in \mathcal{T}$, cost $O(k)$ further enqueue, dequeue and split operations. Note that at any time the storage needed for all these queues is $O(n)$, see Lemma 1. Together with Lemma 11 this observation implies our main result.

Theorem 12 *The weakly greedy flip algorithm is optimal: it computes the tangent visibility graph of a collection of n disjoint convex objects in the plane in $O(n \log n + k)$ time and $O(n)$ working storage, where k is the number of free bitangents.*

5 Topological sweep of the visibility complex

It is worth noting that, translated in "dual space" our algorithm implements an efficient "topological sweep" of the visibility complex, introduced in [7]. We explain this briefly. Recall that the underlying space $|X|$ of the visibility complex X is the quotient space of the space of free rays $\mathcal{F} \times \mathcal{R}$ under the equivalence relation $(p, u) \sim (q, u)$ iff the line segment $[p, q]$ lies in free space $\mathcal{F}$ and the slope of the line (pq) is equal to u modulo π. The topology of $|X|$ induces a natural structure of a 2–dimensional regular cell complex on $|X|$ (see [7]). In particular there is an onto–mapping $b \mapsto |b|$ from the set X_0 of vertices of X to the set of free bitangents. (The preimage of the bitangent $b = [p, q]$ and direction u is the set of rays $(p, u + k\pi)$, $k \in \mathcal{Z}$.) Let x be a face (= vertex, edge or facet) of X. We define $\sup x$ ($\inf x$) to be the ray [1] with maximal (minimal) slope in the closure of x. We turn X into a poset $(X, \prec)$ by taking the transitive closure of the relation $\inf x \prec x \prec \sup x$. (See e.g. [8], chapter 3, for terminology on posets.) A cut of X is a maximal antichain of $(X \setminus X_0, \prec)$. A cut depends only on its subset of edges, and there is exactly one edge per oriented obstacle. We extend $\prec$ to the set $\mathcal{C}$ of cuts by setting $\Phi \prec \Phi'$ iff $E_i \prec E'_i$, for all i, where E_i is the edge in the cut Φ associated with the oriented obstacle i. One can check that if Φ' covers Φ then $\Phi \setminus \Phi'$ and $\Phi' \setminus \Phi$ are composed of 2 edges and 1 facet incident to the same vertex b, i.e. $\sup \Phi \setminus \Phi' = \inf \Phi' \setminus \Phi = b$; we will say that Φ' covers Φ via b. Now the intuitive notion of a topological sweep of X is formally defined as a topological sorting of $(X, \prec)$ or, equivalently, as a maximal chain of the poset $(\mathcal{C}, \prec)$. The following theorem asserts that the flip algorithm realizes a topological sweep of the visibility complex.

[1] By a slight abuse of terminology a point in $|X|$ is still called a ray.

Theorem 13 *Let Φ be a cut of X. Then $\sup \Phi$ is a weakly greedy pseudo–triangulation. Furthermore a vertex b is minimal in $\sup \Phi$ iff Φ is covered by some Φ' via b. In that case $\sup \Phi'$ is obtained from $\sup \Phi$ by flipping b.*

References

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

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

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

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

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

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

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

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

[9] E. Welzl. Constructing the visibility graph for n line segments in $O(n^2)$ time. *Inform. Process. Lett.*, 20:167–171, 1985.