Reaching a Goal with Directional Uncertainty

Mark de Berg Leonidas Guibas Dan Halperin
Mark Overmars Otfried Schwarzkopf Micha Sharir
Monique Teillaud

UU-CS-1994-09
January 1994

Utrecht University
Department of Computer Science

Padualaan 14, P.O. Box 80.089,
3508 TB Utrecht, The Netherlands,
Tel. : + 31 - 30 - 531454

Reaching a Goal with Directional Uncertainty

Mark de Berg   Leonidas Guibas   Dan Halperin
Mark Overmars   Otfried Schwarzkopf   Micha Sharir
Monique Teillaud

Technical Report UU–CS–1994–09
January 1994

Department of Computer Science
Utrecht University
P.O.Box 80.089
3508 TB Utrecht
The Netherlands

ISSN: 0924–3275

Reaching a Goal with Directional Uncertainty*

Mark de Berg[1] Leonidas Guibas[2] Dan Halperin[3]
Mark Overmars[1] Otfried Schwarzkopf[1] Micha Sharir[4]
Monique Teillaud[5]

Abstract

We study two problems related to planar motion planning for robots with imperfect control, where, if the robot starts a linear movement in a certain commanded direction, we only know that its actual movement will be confined in a cone of angle α centered around the specified direction.

First, we consider a single goal region, namely the "region at infinity", and a set of polygonal obstacles, modeled as a set S of n line segments. We are interested in the region $\mathcal{R}_\alpha(S)$ from where we can reach infinity with a directional uncertainty of α. We prove that the maximum complexity of $\mathcal{R}_\alpha(S)$ is $O(n/\alpha^5)$. Second, we consider a collection of k polygonal goal regions of total complexity m, but without any obstacles. Here we prove an $O(k^3 m)$ bound on the complexity of the region from where we can reach a goal region with a directional uncertainty of α. For both situations we also prove lower bounds on the maximum complexity, and we give efficient algorithms for computing the regions.

*This research was supported by the Netherlands' Organization for Scientific Research (NWO) and partially by ESPRIT Basic Research Actions No. 6546 (project *PROMotion*) and No. 7141 (project ALCOM II: *Algorithms and Complexity*). Part of the research was done during the Second Utrecht Workshop on Computational Geometry and its Application, supported by NWO. L.G. acknowledges support by NSF grant CCR-9215219, by a grant from the Stanford SIMA Consortium, and by grants from the Digital Equipment, Mitsubishi, and Toshiba Corporations. D.H. was supported by a Rothschild Postdoctoral Fellowship, by a grant from the Stanford Integrated Manufacturing Association (SIMA), and by NSF/ARPA Research Grant IRI-9306544. Work on this paper by M.S. has been supported by National Science Foundation Grant CCR-91-22103, and by grants from the U.S.-Israeli Binational Science Foundation, the Fund for Basic Research administered by the Israeli Academy of Sciences, and the G.I.F., the German-Israeli Foundation for Scientific Research and Development.

[1]Vakgroep Informatica, Universiteit Utrecht, Postbus 80.089, 3508 TB Utrecht, the Netherlands
[2]Dept. of Computer Science, Stanford University, and DEC Systems Research Center, Palo Alto
[3]Robotics Laboratory, Dept. of Computer Science, Stanford University, Stanford, CA 94305
[4]School of Mathematical Sciences, Tel Aviv University, and Courant Institute of Mathematical Sciences, New York University
[5]INRIA, B.P. 93, 06902 Sophia-Antipolis Cedex, France

1 Introduction and Statement of Result

In this paper we look at regions of the plane defined as the locus of all points for which a cone of angle α can be placed at the point so that certain regions of the plane (the obstacles) are completely avoided, while other regions (the goal regions) are intersected by all rays in the cone. As we explain below, such "visibility" questions arise primarily in robotics, but also in computer graphics and other areas. We give combinatorial bounds and algorithms for the computation of such regions in two special cases.

Many motion planning algorithms in the literature assume that we know the precise geometry of the workspace, and that the robot has precise control over its movements. In practice, however, this will rarely be the case. In most cases, our knowledge of the workspace will be incomplete or erroneous, and the robot can only control its movement imperfectly. As the robot executes a prepared plan to move around the workspace, it will have to deal with uncertainty in the execution of its commanded motions. In many cases it may need to recalibrate its position by sensing the environment or taking equivalent steps.

A motivation for this paper is to understand the effect of uncertainty within a single commanded motion. We have a goal region that the robot wants to reach in one step, while avoiding a certain set of obstacles. We treat the robot as a point—the usual Minkowski sum techniques can be used to reduce to the point case if the robot has finite extent. While we assume perfect knowledge about the scene, our robot does not have full control of its movement: if it starts a linear movement in a certain commanded direction, we only know that its actual movement will be confined in a cone of angle α centered around the specified direction. We are interested in the region from which a certain goal can be reached under these circumstances, and in its complexity and computation.

Such a model was first proposed by Lozano-Pérez, Mason, and Taylor [15] and was further developed in Erdmann's thesis [7] at MIT. For a detailed discussion see the recent book by Latombe [13]. In computational geometry such a model of uncertainty was used for planning compliant motions within a polygonal environment in the works of Briggs [2], Donald [3], and Friedman and others [8, 9]. Most recently Latombe and Lazanas [14] used this model to develop a complete planner for an environment consisting of circular initial, goal, and obstacle regions, as well as circular landmark regions in which the robot has perfect sensing and control.

Similar geometric issues arise in "graphics in flatland", where the goal is to compute global illumination in a two-dimensional scene. Here the goal regions play the role of light sources, and the obstacles are just opaque objects in the environment. In order to obtain a radiosity solution, the environment needs to be meshed, and this meshing needs to be done in accordance with discontinuities in the illumination function. See Heckbert [10, 11].

In the present paper, we consider two special cases of the general problem presented above. In the first situation, we consider a single goal region, namely the "region at infinity", and a set of polygonal obstacles, modeled as a set S of n disjoint line segments. We are interested in the region $\mathcal{R}_\alpha(S)$ from where we can reach infinity with a directional uncertainty of α. We first observe that if the uncertainty angle α is not bounded from below, the complexity of $\mathcal{R}_\alpha(S)$ can be $\Theta(n^4)$. In practice, however, we can assume that α is bounded from below by some constant. Under this condition, we obtain a much better complexity of $O(n/\alpha^5)$. Our proof techniques for this case use recent geometric results of Matoušek et al. [16] and van Kreveld [18] about the arrangements of fat geometric objects. Our result generalizes the case considered by Bhattacharya et al. [1], where the obstacles form a single simple polygon.

In the second situation, we consider a collection of k polygonal goal regions of total complexity m, but without any obstacles. We are again interested in the region from where we can reach some goal region (we do not care which one) within the specified uncertainty. Surprisingly, it turns out that in this case it doesn't help to assume that α is bounded from below, since we can construct an example where the complexity of the region is $\Omega(k^4 + k^2m)$ even for constant α. For this case we prove an upper bound of $O(mk^3)$.

We also show corresponding computational results. For the former problem, our algorithm takes a factor of $O(\log n)$ more than the worst-case combinatorial bound, namely it takes $O((n/\alpha^5)\log n)$. For the second problem we currently only have a rather naive algorithm which runs in $O(k^5 m)$ time.

2 Moving to infinity

In this section we assume that we are given a set S of n line segments with disjoint interiors—we will just call them "disjoint segments" in the sequel—as well as an angle $\alpha > 0$, and we want to find the region $\mathcal{R}_\alpha(S)$ of all points from which we can reach infinity with directional uncertainty α without hitting any obstacle segment in S. Observe that, since the segments are allowed to touch, our setting subsumes that of disjoint simple polygons as obstacles. More formally, let us define an α-cone to be a cone with apex angle α. We assume α to be less than π, and consider α-cones as oriented, so an α-cone has a left ray and a right ray that form an angle of α. We call an α-cone *safe (with respect to S)* if its interior does not intersect any segment in S. A point $x \in \mathbb{E}^2$ is safe if and only if there is a safe α-cone with apex x. Finally, the region $\mathcal{R}_\alpha(S)$ is defined as the locus of all safe points in $\mathbb{E}^2$.

2.1 Combinatorial bounds

In this section we prove bounds on the maximum complexity of the safe region $\mathcal{R}_\alpha(S)$. We will give bounds depending on both n and α, because—due to practical considerations—we are mostly interested in the case where α is a fixed constant. Indeed, for constant α the safe region will be shown to have linear complexity, whereas the best bound that is independent of the value of α is $\Theta(n^4)$, as we show first.

We start with a few general observations. A point on the boundary of $\mathcal{R}_\alpha(S)$ is either on a segment of S or is the apex of an α-cone w that has endpoints p and q of some segments of S on its left and right rays. We say that such an α-cone is *determined* by p and q. The apices of all α-cones determined by two fixed endpoints p and q form two circular arcs, see Figure 1. This implies that the boundary of $\mathcal{R}_\alpha(S)$ is bounded by circular arcs and straight line segments that are pieces of the original segments in S.

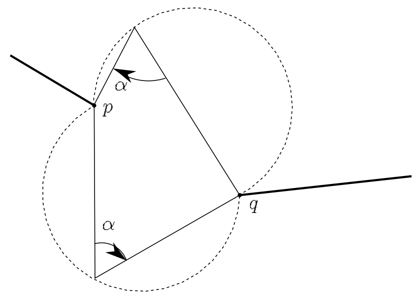

Figure 1:

Theorem 1 *Given a set S of n disjoint line segments, and an angle $\alpha < \pi$. Then the complexity of the region $\mathcal{R}_\alpha(S)$ is bounded by $O(n^4)$. Furthermore, for every n there is a set S of n line segments and an angle $\alpha > 0$ (which decreases with n) such that $\mathcal{R}_\alpha(S)$ has complexity $\Omega(n^4)$.*

Proof: As for the upper bound, we observe that since there are $O(n^2)$ pairs of endpoints, the circular arcs on the boundary of $\mathcal{R}_\alpha(S)$ lie on $O(n^2)$ circles. A vertex of $\mathcal{R}_\alpha(S)$ is an intersection point between two such circles, or between such a circle and a segment in S. It follows that the complexity of $\mathcal{R}_\alpha(S)$ is at most $O(n^4)$.

Figure 2 shows that it is actually possible to achieve this complexity. The example consists of a closed rectangle with two parallel walls at the right side and the bottom side. By poking $n/2$ holes into the right walls we create $\Theta(n^2)$ thin safe regions—one for each pair of holes, one hole in the outer wall and one hole in the inner wall. We poke holes into the bottom walls in a similar way. If the holes are small enough, these regions resemble rays sufficiently to create within the rectangle the equivalent of an arrangement of rays of complexity $\Theta(n^4)$. $\square$

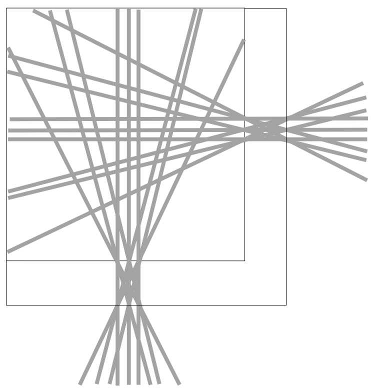

Figure 2: Lower bound example

Note that to realize the lower bound, we have to use a value of α that decreases quite fast when n grows. Therefore, we turn our attention to more useful bounds in terms of α. Especially for the case where safe cones must have an angle that is bounded from below by a constant, we will be able to show a much stronger result.

It turns out to be useful to consider the following directed version of the problem. Let $\vec{u}$ be a direction vector, and let $\mathcal{R}_{\alpha,\vec{u}}(S)$ be the region of all points $x \in \mathbb{E}^2$ such that there is a safe α-cone w with apex x such that the ray with origin x and direction $\vec{u}$ lies in the closure of w. We proceed to analyze the complexity of $\mathcal{R}_{\alpha,\vec{u}}(S)$. We assume without loss of generality that the preferred direction $\vec{u}$ is the upward vertical direction, i.e. the positive y-direction.

Notice that the boundary γ of $\mathcal{R}_{\alpha,\vec{u}}(S)$ is a chain with the property that its intersection with any line with direction $\vec{u}$ is a point or a segment. We will call such a chain *semi-monotone (in direction $\vec{u}$)*. Furthermore, γ consists of circular arcs (determined by two endpoints of S), line segments (pieces of the segments of S), and vertical segments (below an endpoint of a segment of S).

Let P be the set of endpoints of S. We define $\mathcal{R}_{\alpha,\vec{u}}(P)$ analogously to $\mathcal{R}_{\alpha,\vec{u}}(S)$, i.e. $x \in \mathcal{R}_{\alpha,\vec{u}}(P)$ if there is an α-cone w with apex x whose interior does not contain a point of P and such that the ray from x with direction $\vec{u}$ is contained in w.

Lemma 2 *$\mathcal{R}_{\alpha,\vec{u}}(S)$ is the intersection of $\mathcal{R}_{\alpha,\vec{u}}(P)$ with the region above the upper envelope of S.*

Proof: Since P is the set of endpoints of the segments in S we have $\mathcal{R}_{\alpha,\vec{u}}(S) \subset \mathcal{R}_{\alpha,\vec{u}}(P)$. Furthermore, the upward vertical ray from any point below the upper envelope intersects a segment in S, so it cannot be the apex of a safe cone containing the vertical direction.

On the other hand, consider a point x in the intersection of $\mathcal{R}_{\alpha,\vec{u}}(P)$ with the region above the upper envelope of S. Since $x \in \mathcal{R}_{\alpha,\vec{u}}(P)$ there is an α-cone w with apex x that does not contain any endpoint of a segment in S in its interior. Moreover, since x lies above the upper envelope of S and w contains the vertical direction, no segment in S can completely cross w. Hence, w is safe with respect to S. $\square$

Lemma 3 *The complexity of* $\mathcal{R}_{\alpha,\vec{u}}(S)$ *is* $O(n)$.

Proof: Since the boundary of $\mathcal{R}_{\alpha,\vec{u}}(P)$ and the upper envelope of S are both semi-monotone chains, and the latter has complexity $O(n)$, the result follows from Lemma 2 if we can prove that the complexity of $\mathcal{R}_{\alpha,\vec{u}}(P)$ is $O(n)$.

The boundary of $\mathcal{R}_{\alpha,\vec{u}}(P)$ consists of circular arcs and vertical segments. A vertex x of $\mathcal{R}_{\alpha,\vec{u}}(P)$ either lies below a point of P—there are at most $4n$ such vertices, namely two for each of the $2n$ points of P—or is the apex of an α-cone with at least three points of P on its bounding rays. We first count the vertices where there are at least two points of P on the left ray of this α-cone. To this end we observe that the rightmost of these points cannot play this role for more than one vertex: Suppose that there are two α-cones, both having at least two points on their left ray, which share the rightmost point on their left ray. Then one of the two cones must contain the leftmost point on the left ray of the other cone, as illustrated in Figure 3. Thus, there are at most $2n$ such vertices. Vertices with at least two points on the right ray of the corresponding α-cone are counted in the same way. This proves Lemma 3. $\square$

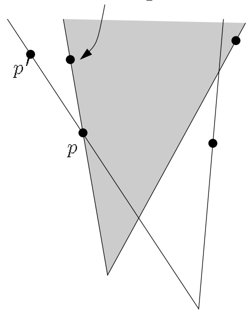

Figure 3: There cannot be two α-cones with common point p

We will exploit this lemma to bound the complexity of $\mathcal{R}_{\alpha}(S)$. We first observe that there is a collection U of $O(1/\alpha)$ different orientations such that

$$\mathcal{R}_{\alpha}(S) = \bigcup_{\vec{u}\in U} \mathcal{R}_{\alpha,\vec{u}}(S).$$

Next we note that any vertex of $\mathcal{R}_{\alpha}(S)$ is a vertex of $\mathcal{R}_{\alpha,\vec{u}}(S) \cup \mathcal{R}_{\alpha,\vec{v}}(S)$ for some pair of $\vec{u}$, $\vec{v}$ in U. We will show that the complexity of such a union $\mathcal{R}_{\alpha,\vec{u}}(S) \cup \mathcal{R}_{\alpha,\vec{v}}(S)$ is $O(n/\alpha^3)$. Since there are $O(1/\alpha^2)$ possible pairs of $\vec{u}$ and $\vec{v}$, this will prove an upper bound of $O(n/\alpha^5)$ on the complexity of $\mathcal{R}_{\alpha}(S)$.

So let us fix two directions $\vec{u}$ and $\vec{v}$, and consider the regions $\mathcal{R}_{\alpha,\vec{u}}(S)$ and $\mathcal{R}_{\alpha,\vec{v}}(S)$. These regions cannot have any long and skinny parts—after all, they are unions of (infinitely many) α-cones, so the value of α gives a lower bound on the "skinniness" of $\mathcal{R}_{\alpha,\vec{u}}(S)$. In fact, this is the concept of *fatness* employed by Matoušek et al. [16] and van Kreveld [18]. They have proven results on the number of holes in the union of fat regions, which can in turn be used to bound the complexity of their union. Unfortunately, these results are only proven for polygonal regions, and our regions are bounded by circular segments. We will circumvent this problem by approximating the circular arcs by line segments, and proving that this does not increase the complexity of the union too much.

Let γ be the boundary of $\mathcal{R}_{\alpha,\vec{u}}(S)$, and μ be the boundary of $\mathcal{R}_{\alpha,\vec{v}}(S)$. γ and μ are semi-monotone with respect to the directions $\vec{u}$ and $\vec{v}$, resp. We partition γ and μ into pieces at their break points. We denote the resulting set of pieces by $\gamma_1, \gamma_2, \ldots$ and $\mu_1, \mu_2, \ldots$, respectively. Note that each γ_i (or μ_i) is a line segment or a circular arc. We will treat all these pieces separately. For a piece γ_i, let γ_i' be the segment connecting the two endpoints of γ_i, and let γ_i'' be the polygonal chain obtained by replacing γ_i by two vertical segments and a horizontal segment through its lowest point, as in Figure 4. (Here, $\vec{u}$ is assumed to be vertical). We define γ' and γ'' to be the union of the pieces γ_i' and the γ_i'', respectively. Let Δ_i^γ be the possibly degenerated trapezoid enclosed between γ_i' and γ_i''. Define μ_j', μ_j'', μ', μ'' and Δ_j^μ in the same way.

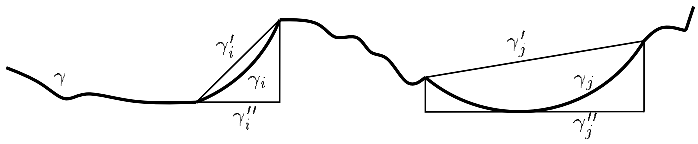

Figure 4: Replacement of circular arcs in γ

Consider now a pair γ_i and μ_j. Because of their simple shape, those two pieces can have at most a constant number of intersections, or, equivalently, can contribute at most a constant number of vertices to the union of $\mathcal{R}_{\alpha,\vec{u}}(S)$ and $\mathcal{R}_{\alpha,\vec{v}}(S)$. To estimate the complexity of $\mathcal{R}_{\alpha,\vec{u}}(S) \cup \mathcal{R}_{\alpha,\vec{v}}(S)$ it is therefore sufficient to bound the number of pairs γ_i, μ_j that intersect.

Consider now a pair γ_i, μ_j that intersect. If μ_j lies completely within the trapezoid Δ_i^γ, then μ_j cannot intersect any other $\gamma_{i'}$. It follows that there are at most $O(n)$ such intersections, and the same reasoning holds for the case that γ_i lies in Δ_j^μ. For all remaining intersecting pairs γ_i, μ_j, there must also be an intersection between two of the curves γ_i', γ_i'', μ_j', and μ_j''. Or, equivalently, for every such pair there is a vertex in $\mathcal{R}'_{\alpha,\vec{u}}(S) \cup \mathcal{R}'_{\alpha,\vec{v}}(S)$, $\mathcal{R}'_{\alpha,\vec{u}}(S) \cup \mathcal{R}''_{\alpha,\vec{v}}(S)$, $\mathcal{R}''_{\alpha,\vec{u}}(S) \cup \mathcal{R}'_{\alpha,\vec{v}}(S)$, or $\mathcal{R}''_{\alpha,\vec{u}}(S) \cup \mathcal{R}''_{\alpha,\vec{v}}(S)$, where $\mathcal{R}'_{\alpha,\vec{u}}(S)$ ($\mathcal{R}'_{\alpha,\vec{v}}(S)$) is the region above γ' (μ') and $\mathcal{R}''_{\alpha,\vec{u}}(S)$ ($\mathcal{R}''_{\alpha,\vec{v}}(S)$) is the region above γ'' (μ''). It is important here that the regions $\mathcal{R}'_{\alpha,\vec{u}}(S)$, etc., are defined to be open; otherwise some intersections can be missed. So, it will be sufficient to prove that the complexity of all these pairwise unions is $O(n/\alpha^3)$.

We now use the *combination lemma* by Edelsbrunner et al. [4]. It states that the complexity of the union of two polygonal regions R and R' is bounded by the complexities of the two components plus the number of holes in $R \cup R'$. It remains to show that the number of holes in the above unions is in $O(n/\alpha^3)$.

To this end we first show that both $\mathcal{R}'_{\alpha,\vec{u}}(S)$ and $\mathcal{R}''_{\alpha,\vec{u}}(S)$ can be covered by $O(n)$ *β-fat triangles*—triangles whose smallest angle is bounded from below by β—with $\beta \geq c\alpha$ for some constant $c > 0$. We use the technique by van Kreveld [18]. He defines a polygon $\mathcal{P}$ to be *δ-wide* if it does not contain a γ-corridor for $\gamma < \delta$; here a γ-corridor is defined as a quadrilateral with vertices v_1, v_2, v_3, v_4 such that v_1, v_2 lie on some edge e of $\mathcal{P}$, v_3, v_4 lie on some edge e', $\angle v_1v_2v_3 = \angle v_2v_3v_4$, $\angle v_3v_4v_1 = \angle v_4v_1v_2$ and $|\overline{v_1v_2}| = |\overline{v_3v_4}| = \frac{1}{\gamma}\max\{|\overline{v_2v_3}|, |\overline{v_4v_1}|\}$. Informally speaking, a γ-corridor is a symmetric trapezoid with vertices on two edges of $\mathcal{P}$ whose width-length ratio is γ. Van Kreveld has proven that any δ-wide polygon can be covered with $O(n)$ $(c'\delta)$-fat triangles for some constant $c' > 0$. Since any corridor in $\mathcal{R}'_{\alpha,\vec{u}}(S)$ must contain an α-cone, its width-length ratio cannot be worse than $\sin\alpha$. It follows that we can cover $\mathcal{R}'_{\alpha,\vec{u}}(S)$ and $\mathcal{R}''_{\alpha,\vec{u}}(S)$ with $O(n)$ β-fat triangles with $\beta \geq c\alpha$ for some constant $c > 0$. Now we can apply a result by Matoušek et al. [16] which states that the union of n β-fat triangles has at most $O(n/\beta^3)$ holes. Applied to our case, this gives us the $O(n/\alpha^3)$ bound on the union of $\mathcal{R}'_{\alpha,\vec{u}}(S)$ and $\mathcal{R}''_{\alpha,\vec{u}}(S)$ we are looking for.

Theorem 4 *Let S be a set of n disjoint line segments in the plane, and let $\alpha < \pi$ be given. The complexity of $\mathcal{R}_\alpha(S)$ is $O(n/\alpha^5)$. Moreover, there is an example of n line segments where the complexity of $\mathcal{R}_\alpha(S)$ is $\Omega(n/\alpha)$.*

Proof: We have proven the upper bound above, so it only remains to give the lower bound example. Figure 5 gives a sketch. It contains $O(1/\alpha)$ obstacle segments, and has complexity $\Omega(1/\alpha^2)$. By combining $O(n\alpha)$ of these gadgets, we obtain an example with n segments and complexity $O(n/\alpha)$. ◻

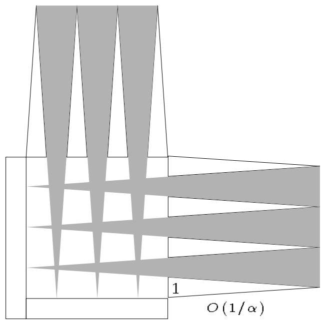

Figure 5: Lower bound construction for given α

2.2 Algorithms

We now describe an algorithm to compute the region $\mathcal{R}_\alpha(S)$ efficiently. We essentially follow the ideas used in the combinatorial proof. We first show how to compute $\mathcal{R}_{\alpha,\vec{u}}(S)$, for a fixed direction $\vec{u}$, and then use a divide and conquer algorithm on the set U of $O(1/\alpha)$ directions to compute $\mathcal{R}_\alpha(S)$.

Lemma 5 *Given a set S of n disjoint line segments, a direction vector $\vec{u}$, and an angle $\alpha > 0$, the region $\mathcal{R}_{\alpha,\vec{u}}(S)$ can be computed in time $O(n \log n)$.*

Proof: Assume without loss of generality that $\vec{u}$ is the upward vertical direction. As we observed earlier, the region $\mathcal{R}_{\alpha,\vec{u}}(S)$ can be found by intersecting $\mathcal{R}_{\alpha,\vec{u}}(P)$—where P is the set of endpoints of S—with the region above the upper envelope of S. Suppose that we have computed $\mathcal{R}_{\alpha,\vec{u}}(P)$. Then we can compute the upper envelope and its intersection with $\mathcal{R}_{\alpha,\vec{u}}(P)$ in $O(n \log n)$ time using a plane sweep. So it remains to compute $\mathcal{R}_{\alpha,\vec{u}}(P)$.

As we have seen before, the boundary γ of $\mathcal{R}_{\alpha,\vec{u}}(P)$ is a semi-monotone curve consisting of circular arcs and vertical line segments. We construct the chain γ from left to right. Assume that we are at a certain breakpoint x on γ, i.e. we have constructed the part of γ to the left of x, and we want to determine how to continue from here. We distinguish two cases.

The first case is where x lies directly below some point p of P and the previous breakpoint does not lie below the same point p. Now there will be a vertical segment on γ. To find the other endpoint of this segment, we have to find the maximal safe α-cone that has a vertical left ray with p on it. This cone can be found by determining the first point of P that is hit when we shift an α-cone, whose apex has the same x-coordinate as p and whose left ray is vertical, downward from infinity. In other words, if we denote the subset of P of points with x-coordinate larger than p by $P_{>p}$, we want to find the point of $P_{>p}$ extreme in some fixed direction. All these points can be precomputed in time $O(n \log n)$, by constructing the upper convex hull of P incrementally from right to left, maintaining the point in the current hull extreme in the requested direction.

The second case is when we are at a certain breakpoint x on γ, and we know that the part of γ directly to the right of x is a circular arc, determined by two points p and q. Let w be the α-cone at x determined by p and q, see Figure 6. To find the next vertex of γ, we have to find the first point of P intersected by w when we "rotate" it, while keeping contact with p and q, until the right ray of w gets vertical. There are two candidates for this point. The first one is the first point hit by a vertical ray with origin p rotated leftward, the second one is the first point in the vertical slab between x and q hit by the segment $\overline{qx'}$ when x' moves along the circle through p, q and x, starting at x. We will determine both such points, and choose the one involving the smaller angle of rotation. If this angle is larger than the angle which rotates the right ray of w into vertical direction, we move x to a point below q, and continue as in the first case above. If the rotation of w hits a point r we are again in the second case; notice that we know the two points which define the next arc (either p and r, or q and r).

It remains to implement the operations of finding the first point hit by a ray rotated leftward around p, and by the segment $\overline{qx'}$ when x' moves as described above. The first operation is easy: the first point hit when we rotate a vertical ray around p is just the left neighbor of p on the convex hull of the subset $P_{\leq p}$ of points in P left of and including p. All these points can be precomputed by computing the upper convex hull of P incrementally from left to right, and storing for each point p its left neighbor at the time of its insertion. All this takes time $O(n \log n)$.

The second operation is a bit more tricky. What we do is determine the first point in the set $P_{>x}$ that is hit by a ray ρ_q rotated counterclockwise around q, starting at the vertically upward position. Notice that because the cone touching p and q with apex x is empty, the ray ρ_q cannot hit any point in $P_{>x}$ before it reaches x. However, when we rotate ρ_q further it might hit a point r' that would not have been hit by the segment $\overline{qx'}$. Fortunately, every point in P can be found at most once in this fashion, and we can charge the cost of this rotation query with ρ_q to r'. So we perform another rotation query with the ray ρ_q, this time in the set $P_{>r'}$. The process is repeated until finally a point is found which is also hit by the segment $\overline{qx'}$, or ρ_q becomes vertical. So it remains to implement the following operation: given a point $q \in P$ and a point x, find the first point in the set $P_{>x}$ that is hit by the ray ρ_q when rotated counterclockwise around q, starting at the vertically upward position. Again, this is the left neighbor of q in the convex hull of $P_{>x}$. We can precompute these points by computing the convex hull of P incrementally from right to left, and storing with each point q on the convex hull the moments when its left neighbor changes. Since there are only $O(n)$ changes during the construction, this takes only $O(n)$ time and space. The final structure (consisting of an array for every point q that stores its intervals and the corresponding hit points) is used to answer the rotational queries of our algorithm in time $O(\log n)$. It follows that the total running time of the algorithm is in $O(n \log n)$. $\square$

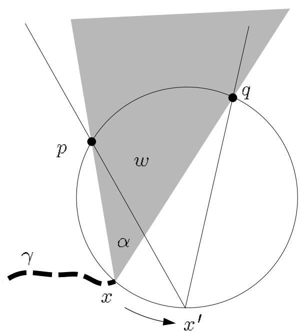

Figure 6: Next vertex by rotating the α-cone

It is not difficult to verify that if the points are given sorted from left to right, then $\mathcal{R}_{\alpha,\vec{u}}(P)$ can actually be computed in linear time.

Theorem 6 *Let S be a set of n disjoint line segments in the plane, and let $\alpha < \pi$ be given. Then $\mathcal{R}_\alpha(S)$ can be computed in time $O((n/\alpha^5)\log n)$.*

Proof: We use a divide and conquer algorithm on the set U of $O(1/\alpha)$ directions. The merging step can be done in time $O((n+K)\log n)$ by a standard plane sweep algorithm, where K is the complexity of the merged region. If we denote the number of directions in U by s then we have $K = O(s^2 \frac{n}{\alpha^3}\log n)$, by the combinatorial results of the previous section. We thus obtain the following recursion for $T(s)$, the time for computing the union of $\mathcal{R}_{\alpha,\vec{u}}(S)$ for s different directions $\vec{u}$.

$$\begin{aligned} T(1) &= O(n\log n) \\ T(s) &= 2T(s/2) + O(s^2 \frac{n}{\alpha^3}\log n), \end{aligned}$$

which solves to $T(s) = O(s^2 \frac{n}{\alpha^3}\log n)$. Substituting $s = O(1/\alpha)$ gives the claimed time bound. $\square$

3 Multiple goal regions

In this second part of the paper we study the following problem. We are given a family $\mathcal{B}$ of k pairwise disjoint polygonal goal regions with a total complexity m, and we are interested in the region $\mathcal{R}_\alpha(\mathcal{B})$ from where some goal in $\mathcal{B}$ can be reached with directional uncertainty $\alpha > 0$. More formally, we will say that an α-cone w with apex x is *safe* if and only if every ray with origin x that lies in w intersects an element of $\mathcal{B}$. We call a point $x \in \mathbb{E}^2$ *safe* if there is a safe α-cone w with apex x, and define $\mathcal{R}_\alpha(\mathcal{B})$ as the region of all safe points. We will prove bounds on the maximum complexity of the region $\mathcal{R}_\alpha(\mathcal{B})$, and then present an efficient algorithm to construct the region.

3.1 Combinatorial bounds

We first observe that we can assume that the polygons in $\mathcal{B}$ are convex if $\alpha < \pi$. This is true because for $\alpha < \pi$, we can always reach a polygon B from any point within its convex hull, and a ray with origin outside the convex hull of B intersects B exactly if it intersects its convex hull, see Figure 7. Notice that the convex hulls of a set of disjoint polygons are not necessarily disjoint. However, if two or more of the convex hulls intersect then we can repeat the above argument, and replace them by the convex hull of their union. This process continues until we are left with a set of disjoint convex polygons. Notice that every vertex of the remaining polygons must be a vertex of one of the original polygons, so the total complexity of the polygons has not increased.

Let us start by considering a single convex goal polygon B with m vertices. The region $\mathcal{R}_\alpha(\{B\})$ is a flower-shaped region, bounded by circular arcs. Let γ be the closed boundary curve of $\mathcal{R}_\alpha(\{B\})$. For a point x on γ, there is an α-cone w whose boundary rays are tangent to B. For vertices of γ, one of the boundary rays is flush with an edge of B. To bound the number of vertices of γ we thus have to bound the number of edge-vertex pairs such that there is an α-cone with one of its rays containing the edge, and the other ray being tangent to B at the vertex. Observe that each edge of B defines at most two such pairs: an edge with orientation θ forms a pair exactly with the two extreme vertices of B in the directions orthogonal to $\theta - \alpha$ and $\theta + \alpha$. Consequently, the complexity of $\mathcal{R}_\alpha(\{B\})$ is in $O(m)$. The example of a regular convex m-gon shows that this bound can actually be achieved. It is not difficult to compute $\mathcal{R}_\alpha(\{B\})$ in linear time: the relevant edge-vertex pairs can easily be computed after merging the ordered list of all orientations of edges of B with the same list with α added to the orientations.

The above discussion is summarized in the following lemma.

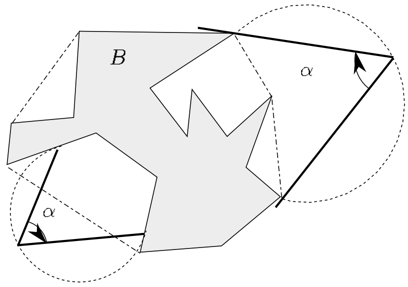

Figure 7: We can assume all polygons to be convex

Lemma 7 *The maximum complexity of the region* $\mathcal{R}_\alpha(\{B\})$ *of a convex polygon* B *with* m *vertices is* $\Theta(m)$. *Moreover,* $\mathcal{R}_\alpha(\{B\})$ *can be computed in* $O(m)$ *time.*

We now turn our attention to the case where we have a family $\mathcal{B} = \{B_1, \ldots, B_k\}$ of k disjoint convex goal regions. Let m_i denote the number of vertices of B_i and let $m = \sum_{i=1}^{k} m_i$ be the total number of vertices. Notice that is is not sufficient to simply take the union of the regions $\mathcal{R}_\alpha(\{B_i\})$, because some points may not have an α-cone that is safe by any single goal region but only an α-cone safe due to several goal regions.

Consider a (circular) piece of the boundary of $\mathcal{R}_\alpha(\mathcal{B})$ which is defined by more than one goal region. There can be more than two goal regions which are needed to make sure that points on this boundary piece have a safe α-cone. However, for points on the boundary of $\mathcal{R}_\alpha(\mathcal{B})$ there is an α-cone that touches only two of them, each in a vertex. So the question becomes: how many pairs of vertices, one from B_i and one from B_j, can there be such that there is an α-cone touching B_i at one vertex and touching B_j at the other vertex? Now we note that such a pair of vertices also defines an α-cone which touches the convex hull of B_i and B_j in two points (namely, in the two vertices). This convex hull has at most $m_i + m_j$ vertices, so by Lemma 7 there are only $O(m_i + m_j)$ such pairs. Summing over all pairs of polygons, we obtain

$$O\Big(\sum_{1 \le i \le j \le k} (m_i + m_j)\Big) = O\Big(\sum_{i=1}^{k}\sum_{j=1}^{k} m_j\Big) = O\big(\sum_{i=1}^{k} m\big) = O(km).$$

It follows that there are only $O(km)$ possible pairs of vertices that can determine an arc of the boundary of $\mathcal{R}_\alpha(\mathcal{B})$.

However, the complexity of $\mathcal{R}_\alpha(\mathcal{B})$ can be a lot higher, because the circular arc defined by a pair of vertices can appear in several pieces on the boundary of $\mathcal{R}_\alpha(\mathcal{B})$. To see what happens it is useful to go back to the case of one goal region B, and to take a somewhat different view on $\mathcal{R}_\alpha(\{B\})$. For every pair p, q of vertices of B let $C(p,q)$ be the region $\mathcal{R}_\alpha(\{\overline{pq}\})$. $C(p,q)$ is the union of two discs as in Figure 8. Clearly, $\mathcal{R}_\alpha(\{B\})$ is just the union of all $C(p,q)$, for all pairs of p and q. Lemma 7 tells us that only a linear number of pairs is relevant.

Now we return to the case of multiple goal regions. Here we have $O(km)$ pairs (p,q) that define a region $C(p,q)$ which is relevant. The complication is that for vertices p, q of different polygons the whole region $C(p,q)$ is not necessarily contained in $\mathcal{R}_\alpha(\mathcal{B})$: we know that for points in $C(p,q)$ there is an α-cone whose bounding rays intersect two of the goal regions but this α-cone need not be safe.

To obtain this extra information we consider the arrangement $\mathcal{A}(L)$ formed by the set L of lines tangent to two polygons in $\mathcal{B}$. Since there are $O(k^2)$ such lines, the arrangement $\mathcal{A}(L)$ consists of $O(k^4)$ cells. Consider a cell c of $\mathcal{A}(L)$. With each cell c of $\mathcal{A}(L)$ we associate a *visibility cycle* $\mathcal{V}_c$, defined as the circularly ordered list of visible polygons intersected by a ray rotating clockwise around any given point in the cell. Whenever a ray does not intersect any polygon, the

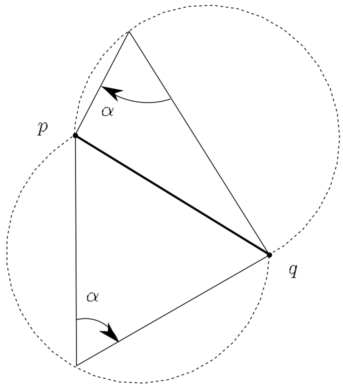

Figure 8: $C(p, q) = \mathcal{R}_\alpha(\{\overline{pq}\})$

corresponding element in the cycle is denoted as ∞. Each visibility cycle contains $O(k)$ elements, and it consists of several connected components, separated by ∞. See Figure 9, which shows the visibility cycle for every cell of the arrangements of tangents. Observe that the visibility cycle of a cell c of $\mathcal{A}(L)$ is well defined, that is, $\mathcal{V}_c$ does not depend on which point in c is chosen. But then it readily follows that within every cell c of $\mathcal{A}(L)$, the region $\mathcal{R}_\alpha(\mathcal{B})$ is equal to the union of the regions $C(p, q)$, where the union is taken over all pairs of p and q that come from polygons in the same connected component in $\mathcal{V}_c$. Thus within every cell the region is equal to the union of $O(km)$ discs, which has $O(km)$ complexity [12]. Since $\mathcal{A}(L)$ has $O(k^4)$ cells the total complexity of $\mathcal{R}_\alpha(\mathcal{B})$ is $O(k^5 m)$. However, it is possible to do better if we observe that a disc is interesting in a certain cell of $\mathcal{A}(L)$ only if its boundary intersects the cell—otherwise the disc either makes the whole cell part of $\mathcal{R}_\alpha(\mathcal{B})$, or it cannot participate in the complexity within this cell at all. A circle can intersect a line at most twice, and hence can intersect at most $O(k^2)$ cells of our arrangement. Since we have $O(km)$ discs, we find that the number of interesting cell-disc pairs is only $O(k^3 m)$. It follows that the total complexity of $\mathcal{R}_\alpha(\mathcal{B})$ is at most $O(k^3 m)$.

Theorem 8 *Given a family $\mathcal{B}$ of k polygons of total complexity m and an angle $0 < \alpha < \pi$, the total complexity of $\mathcal{R}_\alpha(\mathcal{B})$ is at most $O(k^3 m)$. There is an example of k goal polygons with total complexity m such that the complexity of $\mathcal{R}_\alpha(\mathcal{B})$ is $\Omega(k^4 + k^2 m)$.*

Proof: It only remains to prove the lower bound. Consider Figure 10. If the horizontal segment B_1 is a single line segment, then any point within the shaded region can see B_1 under an angle of α. If, however, we poke a little hole at the middle of B_1, then the shaded region is too far away from the two resulting pieces to see them under an angle of α. This leads to the following construction: we use two parallel line segments B_1 and B_2, and poke k point holes very close to the midpoints of both segments.

This generates k^2 lines on which it is possible to "look through" a pair of holes in B_1 and B_2. It follows that a point in the intersection of such a line with the shaded region is not in $\mathcal{R}_\alpha(\mathcal{B})$. From all other points of the shaded region, however, B_1 appears as a solid segment—there is no visible hole—so all these points belong to $\mathcal{R}_\alpha(\mathcal{B})$. It is possible to arrange the k^2 lines in such a way that they generate an arrangement of complexity $\Omega(k^4)$ within the shaded region, proving a lower bound of $\Omega(k^4)$ on the complexity of $\mathcal{R}_\alpha(\mathcal{B})$.

Consider now Figure 11. We construct a convex m-gon which can be seen from m points on the line ℓ under an angle of α. If we move along the line ℓ, we enter and leave $\mathcal{R}_\alpha(\{B\})$ m times. Now we replace ℓ by a bundle of k^2 lines constructed as in Figure 10. If these lines are sufficiently close to ℓ, we still have the property that if we move along any of these lines, we enter and leave $\mathcal{R}_\alpha(\{B\})$ m times. We furthermore make sure that the interesting part of ℓ in Figure 11 lies in the shaded region of Figure 10. We now have the situation that the whole shaded region belongs

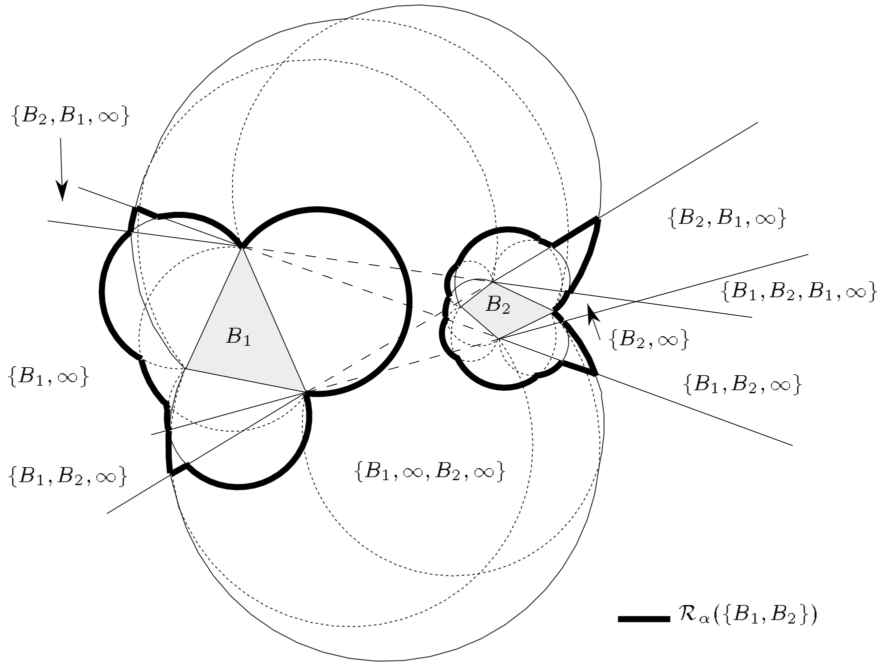

Figure 9: The arrangement of tangents

to $\mathcal{R}_\alpha(\mathcal{B})$, but contains k^2 lines. Every such line contains m pieces that do not belong to $\mathcal{R}_\alpha(\mathcal{B})$. The final result is a region belonging to $\mathcal{R}_\alpha(\mathcal{B})$, but containing $\Omega(k^2 m)$ line segments that do not. This proves the second term of the lower bound. $\square$

3.2 Algorithms

At present, we only have a rather naive algorithm for computing the region $\mathcal{R}_\alpha(\mathcal{B})$. Below we sketch this algorithm briefly, and we indicate the difficulties in obtaining a more efficient algorithm.

We assume that $\mathcal{B}$ consists of disjoint convex polygons. Recall from the previous subsection that $\mathcal{R}_\alpha(\mathcal{B})$ is contained in the union of $O(km)$ discs $C(p, q)$, where p and q are vertices of two polygons B_i and B_j. We start by computing all the relevant discs. To this end we compute the convex hull $C_{i,j}$ of every pair B_i, B_j of polygons, and we compute the region $\mathcal{R}_\alpha(\{C_{i,j}\})$ according to Lemma 7. Together with the regions $\mathcal{R}_\alpha(\{B_i\})$ this will give us all the discs $C(p, q)$ that we have to consider. This takes $O(km)$ time. Next we compute the set L of lines tangent to two polygons; it is straightforward to do this in $O(km)$ time. We then construct the arrangement $\mathcal{A}(L)$ in $O(k^4)$ time [5]. So the first stage of the algorithm takes $O(k^4 + km)$ time in total.

The second stage of the algorithm is as follows. For every cell c in $\mathcal{A}(L)$ we compute its visibility cycle $\mathcal{V}_c$. This can easily be done in $O(k \log k + m)$ time as follows. Pick some point x in the cell, replace every B_i by a suitable segment s_i (such that the view of B_i from x is the same as the view of s_i) and compute the visibility polygon of x in the resulting set of segments [6]. Next, compute in $O(k^2)$ time a two-dimensional array A_c such that $A_c[i, j]$ is true if and only if B_i and B_j are in the same connected component of the visibility cycle $\mathcal{V}_c$. We then test for every disc $C(p, q)$ if it intersects cell c and, if so, if the two polygons B_i and B_j containing the vertices p and q are in the same connected component of $\mathcal{V}_c$. For one disc this can be done in $O(|c|)$ time, where $|c|$ is the complexity of cell c. The total time for the second stage is bounded by $O(\sum_c O(k^2 + |c|km) = O(k^5 m)$.

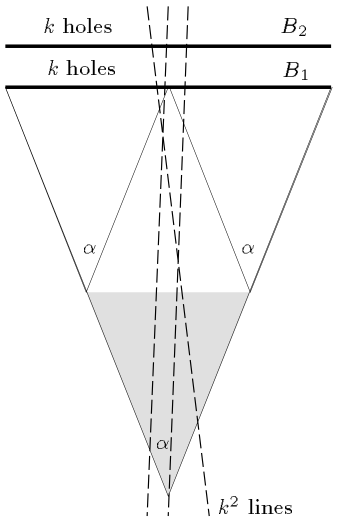

Figure 10: Lower bound of $\Omega(k^4)$

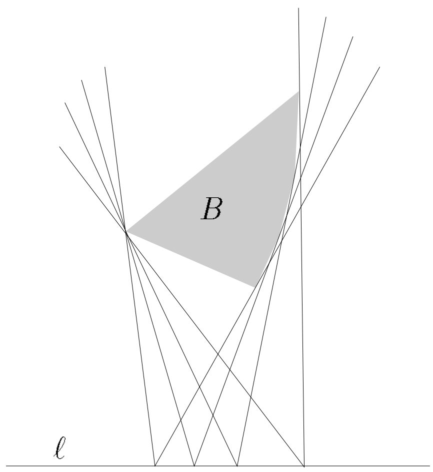

Figure 11: Lower bound of $\Omega(k^2 m)$

Now we have for every cell c of $\mathcal{A}(L)$ a list D_c of the relevant discs. If there is a relevant disc that completely contain c then we add the whole cell to $\mathcal{R}_\alpha(S)$. Otherwise, we compute the union of the discs inside c, which can be done in $O((|c|+|D_c|)\log(|c|+|D_c|))$ randomized time [17]. From the results of the previous subsection we know that $\sum_c |D_c| = O(k^3 m)$. So this takes $O(k^3 m \log m)$ time in total.

We conclude with the following theorem.

Theorem 9 *Given a family $\mathcal{B}$ of k polygons of total complexity m and an angle $0 < \alpha < \pi$, the region $\mathcal{R}_\alpha(\mathcal{B})$ can be computed in $O(k^5 m + k^3 m \log m)$ randomized time.*

Notice that the running time of our algorithm is an $O(k^2)$ factor off the combinatorial bound that we have shown. The problem in obtaining a better running time is the following. Recall that the improved combinatorial bound was obtained by observing that a disc can only influence the complexity of $\mathcal{R}_\alpha(S)$ inside a cell c if its boundary intersects c. We know an efficient algorithm that determines for each disc $C(p,q)$ the cells c where it is relevant (namely where the defining polygons are in the same connected component of $\mathcal{V}_c$) and that are intersected by its boundary. However, we have not yet been able to develop an algorithm that decides for each cell whether it is completely contained in a relevant disc.

4 Conclusion and Extensions

We have studied combinatorial and algorithmic aspects of motion planning for robots with imperfect control, where, if the robot starts a linear movement in a certain commanded direction, we only know that its actual movement will be confined to a cone of angle α around the specified direction. We have studied the case where we have a set of obstacle line segments and we are interested in the locus of all points from where infinity can be reached with directional uncertainty α. We also studied the case where there are no obstacles, but we have a set of polygonal goal regions that we want to reach.

A number of questions is left open. First of all, it would be nice to tighten the gaps in our combinatorial bounds. A related question is the following: We have proven the upper bound in Theorem 4 by first approximating the region by a polygonal region, and then employing bounds from the literature for the union of fat triangles. It would be much nicer if we had tools that allowed us to directly bound the complexity of the union of fat objects with curved boundaries. In that case we could argue that the final region is the union of $O(1/\alpha)$ α-wide "curved polygons", and could probably obtain better bounds than in the present paper.

On the algorithmic side there are also some open problems. For example, the algorithm that we gave for the second problem is not as close to the combinatorial upper bound as we would like it to be. Another challenge is to design an output-sensitive algorithm for computing the region, as the region will not have the worst-case complexity in many practical situations.

Finally, it would be interesting to try to combine our results to obtain results for the general setting, where we have both goal regions and obstacles.

References

[1] B. Bhattacharya, D. G. Kirkpatrick, and G. T. Toussaint. Determining sector visibility of a polygon. In *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, pages 247–254, 1989.

[2] A. J. Briggs. An efficient algorithm for one-step planar compliant motion planning with uncertainty. In *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, pages 187–196, 1989.

[3] B. R. Donald. The complexity of planar compliant motion planning under uncertainty. *Algorithmica*, 5:353–382, 1990.

[4] H. Edelsbrunner, L. J. Guibas, and M. Sharir. The complexity and construction of many faces in arrangements of lines and of segments. *Discrete Comput. Geom.*, 5:161–196, 1990.

[5] H. Edelsbrunner, J. O'Rourke, and R. Seidel. Constructing arrangements of lines and hyperplanes with applications. In *Proc. 24th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 93–91, 1983.

[6] H. Edelsbrunner, M. H. Overmars, and D. Wood. Graphics in Flatland: A case study. In F. P. Preparata, editor, *Computational Geometry*, volume 1 of *Advances in Computing Research*, pages 35–59. JAI Press, London, England, 1983.

[7] M. Erdmann. On motion planning with uncertainty. Technical Report 810, AI Laboratory, MIT, 1984.

[8] J. Friedman, J. Hershberger, and J. Snoeyink. Compliant motion in a simple polygon. In *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, pages 175–186, 1989.

[9] J. Friedman, J. Hershberger, and J. Snoeyink. Input-sensitive compliant motion in the plane. In *Proc. 2nd Scand. Workshop Algorithm Theory*, volume 447 of *Lecture Notes in Computer Science*, pages 225–237. Springer-Verlag, 1990.

[10] P. S. Heckbert. *Simulating Global Illumination Using Adaptive Meshing.* Ph. d. dissertation, University of California, Berkeley, June 1991.

[11] P. S. Heckbert. Discontinuity meshing for radiosity. In *Third Eurographics Workshop on Rendering*, pages 203–216, 1992.

[12] K. Kedem, R. Livne, J. Pach, and M. Sharir. On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete Comput. Geom.*, 1:59–71, 1986.

[13] J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, 1991.

[14] A. Lazanas and J.-C. Latombe. Landmark-based robot navigation. Submitted to Algorithmica.

[15] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor. Automatic synthesis of fine-motion strategies for robots. *Internat. J. Robotics Research*, 3:3–24, 1984.

[16] J. Matoušek, N. Miller, J. Pach, M. Sharir, S. Sifrony, and E. Welzl. Fat triangles determine linearly many holes. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 49–58, 1991.

[17] N. Miller and M. Sharir. Efficient randomized algorithm for constructing the union of fat triangles and of pseudodiscs. Unpublished Manuscript, 1993.

[18] Marc van Kreveld. On fat partitioning, fat covering, and the union size of polygons. In *Proc. 3rd Workshop Algorithms Data Struct.*, volume 709 of *Lecture Notes in Computer Science*, pages 452–463, 1993.