Algorithmica (1992) 8: 321–342

Algorithmica
© 1992 Springer-Verlag New York Inc.

Probing a Scene of Nonconvex Polyhedra[1]

J. D. Boissonnat[2] and M. Yvinec[3]

Abstract. We show, in this paper, how the exact shapes of a class of polyhedral scenes can be computed by means of a simple sensory device issuing probes. A scene in this class consists of disjoint polyhedra with no collinear edges, no coplanar faces, and such that no edge is contained in the supporting plane of a nonincident face. The basic step of our method is a strategy for probing a single simple polygon with no collinear edges. When each probe outcome consists of a contact point and the normal to the object at the point, we present a strategy that allows us to compute the exact shape of a simple polygon with no collinear edges by means of at most $3n - 3$ probes, where n is the number of edges of the polygon. This is optimal in the worst case. This strategy can be extended to probe a family of disjoint polygons. It can also be applied in planar sections of a scene of polyhedra of the class above to find out, in turn, each edge of the scene. If the scene consists of k polyhedra with altogether n faces and m edges, we show that $\frac{10}{3}n(m + k) - 2m - 3k$ probes are sufficient to compute the exact shapes of the polyhedra.

Key Words. Computational geometry, Geometric probing, Polyhedral scenes.

1. Introduction. Given a simple polyhedron or a family of simple nonintersecting polyhedra, the *probing problem* consists in determining the shapes of the polyhedra by a small set of simple measurements. A variety of subproblems can be distinguished, depending on the model of the sensor and on the constraints on the type of the objects to be probed.

This problem was first studied by Cole and Yap [4], who showed that the shape of a convex polygon with n edges can be determined with no more than $3n$ "finger" probes (i.e., each probe response consists of the coordinates of a "contact point" on the boundary of the object); later, Bernstein [3] improved on this result in the case where the polygon is restricted to a finite set. Dobkin *et al.* [5] have considered the case of convex polytopes in multidimensional space, other probe models, and also probes with errors. A work of synthesis of the field of geometric probing as well as a collection of new results can be found in Skiena's Ph.D. thesis [7].

This paper extends the results of [2], recalled in Section 2, where it is shown how we can probe a large class of *nonconvex* polygons, namely the class of simple polygons with no collinear edges. In order to study such complex objects, we use probes that are more powerful than simple finger probes: our probes answer not only with a contact point but also with the normal to the object at that point.

[1] This work has been supported in part by the ESPRIT Basic Research Action No. 3075 (ALCOM).
[2] INRIA, Centre de Sophia Antipolis, 2004 Route des Lucioles, 06565 Valbonne, France.
[3] LIENS, URA CNRS 1327, Ecole Normale Supérieure, 45 rue d'Ulm, 75230 Paris, France.

Received November 6, 1989; revised March 15, 1990. Communicated by Bernard Chazelle.

Moreover, we use an additional information, called *a ray*, which is generally available with the outcome of each probe. A ray is defined as a half-line or, more generally, as any semi-infinite curve which has the measured point as its origin and which does not intersect the interior of the objects—as does an optical ray, for example. It has been shown [1] that, given a set of contact points belonging to the boundary of a single object, the rays induce a total order on the set of points that coincides with the natural order of the points along the boundary of the object. Our method relies heavily on this property and a related lemma that we recall in Section 2.1. The method is subsequently extended to deal with multiple objects in a plane (Section 3) and three-dimensional objects (Section 4).

2. The Basic Planar Probing Algorithm

2.1. Description of the Probe Model and Preliminaries. We show, in this section, how the exact shape of a simple polygon **C** can be computed by probing in the plane of **C**. In the following we denote by n the number of edges of **C**. It is important to realize that n is *a priori* unknown and will be discovered at the same time as the exact shape of the object.

Our probe model is the following. We probe along a half-line, called the *probe path*, whose origin is some point o_i of the plane. When the probe is issued, the probing device responds with the first point p_i, called the *contact point*, where the probe path encounters the boundary of **C** and also gives the normal n_i to **C** at p_i when it is defined. The sensory device is supposed to be able to detect when p_i is a vertex of **C**, in which case the object responds with two normals instead of one, namely, the normals to the edges incident to p_i. An example of such a device may be a finger with a tactile sensor at its tip.

In addition, in order to avoid unrealistic probes, we assume that when the probe path contains an edge of **C** (such a probe is called a *tangent* probe), no contact point on this edge is reported: the device misses the edge. It should be noticed that, in Cole and Yap's model [4], a tangent point returns the first vertex it encounters. This leads to an algorithm that is simpler than the one presented here but highly unreliable if an actual probing device is to be used.

The above probe model does not guarantee that any probing problem is solvable in a finite number of steps. To ensure this, two mild conditions are needed:

CONDITION 1. The oriented supporting lines of the edges of **C** are all distinct.[4] Notice that two supporting lines may be identical if their orientations are opposite.

CONDITION 2. A point t of the object (on the boundary or in the interior of polygon **C**), called the *target point*, is given.

[4] **C** is supposed to be oriented counterclockwise and the edges and their supporting lines accordingly.

These two conditions are made to ensure that the probing problem is solvable in a finite number of steps. Indeed, without the first condition, a small detail of the object may still have been missed after any finite number of probes. The second condition allows us to isolate the problem of discovering the shape of the object from the problem of locating it within the workspace. Without this condition, we have no idea where **C** is located and an unbounded number of probes can be required to find it.

Our probing strategy is based on the use of the total order induced on the set of contact points by the set of probe paths. In order to make use of the results of [1], each new probe is chosen so that the outcoming contact point p_i can be associated with a semi-infinite curve l_i, called a *ray*, that ends at p_i and is known not to intersect the interior of the object. This is achieved as follows. As previously mentioned, each probe is associated with an origin o_i and a contact point p_i. The line segment $o_i p_i$ connecting these two points (a portion of the probing path) is called the *probe segment* of the probe. The origin o_i of a new probe path is chosen to be either a point at infinity or to belong to a previous probe segment. In the former case, ray l_i is identical to the new semi-infinite probe segment; in the latter case, ray l_i is the concatenation of the current probe segment $o_i p_i$ with the semi-infinite prefix made of portions of previous probe segments and ending at point o_i. In the following we consider that a probe outcome, noted $\varpi_i = (p_i, n_i, l_i)$, includes three components: the contact point p_i, the normal n_i to the boundary of **C** at p_i, and the semi-infinite ray l_i ending at p_i.

Let P be a set of contact points and let L be the set of corresponding rays. We first recall a few facts (proved in [1]). The set of rays L induces, on P, a total cyclic order that corresponds to the natural order of the points of P along the boundary of the probed object. The following lemma is a necessary and sufficient condition for two contact points p_i and p_j of P to be consecutive in that order.

Let C be any simple curve joining the points of P without intersecting the rays of L (except at the points of P). In particular, in this section, we can take this curve to be the unknown boundary of the probed object. The curve C is considered to be oriented so that the rays of L lie on the right-hand side of C. Let $C_{i,j}$ be the portion of C joining p_i to p_j. Portion $C_{i,j}$, together with the rays l_i and l_j measuring respectively the points p_i and p_j, partitions the plane into several regions. Let $W_{i,j}$ be the union of the regions that do not contain p_i nor p_j ($W_{i,j}$ may be empty). Among the two regions containing p_i and p_j on their boundary, let $H_{i,j}$ be the region to the right of $C_{i,j}$ (see Figure 1).

LEMMA 1. *Two points p_i and p_j of P are consecutive in the order induced by L if and only if the region $H_{i,j}$, considered as a closed region including its boundary, contains no point of P, except p_i and p_j.*

2.2. The Basic Probing Algorithm. In this section we present a probing strategy that computes the exact shape of an n-sided simple polygon with no collinear edges by means of at most $3n - 3$ probes.

Given a probe outcome $\varpi_i = (p_i, n_i, l_i)$, we call the line D_i, normal to n_i and passing through p_i, the *supporting line* of ϖ_i. When necessary, D_i is considered

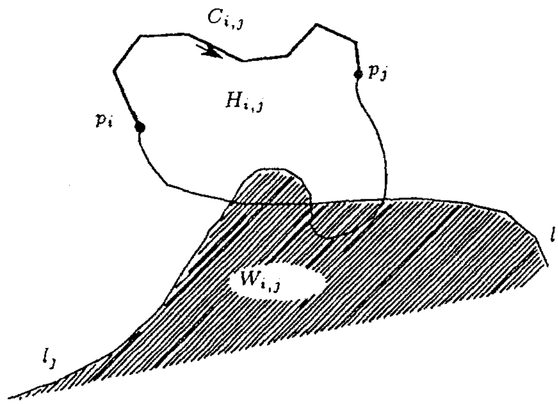

Fig. 1. For the definition of $W_{i,j}$ and $H_{i,j}$.

to be oriented so as to let l_i lie on its right-hand side (and therefore the interior of the object on its left-hand side) in the neighborhood of p_i. When a probe outcome includes a contact point p_i that belongs to the edge e_i of **C**, we say that the edge e_i has been *discovered*. At that time, this edge is not completely known because its endpoints have not yet been found out.

The initialization step of the algorithm performs the first three probes as follows. The first two probes are issued along straight line rays with opposite directions and both passing through the target. Let D_1 and D_2 denote the two supporting lines of the two corresponding probe outcomes ϖ_1 and ϖ_2 and let $I = D_1 \cap D_2$ be the intersection point (possibly at infinity) of these two lines. The third probe is performed along a directed straight line passing through the target point and I and directed in such a way that the target point is reached before I. The three corresponding contact points p_1, p_2, p_3 belong to three distinct edges of **C**.

Let us now describe a step of the core of the algorithm. At a given stage of the algorithm some edges have been discovered. The algorithm maintains a list of contact points $\mathscr{C}$, sorted according to the ray order (in the following the indices refer to that order). The intersection I between the supporting lines D_i and D_{i+1} of two successive contact points is called a *corner* and is a potential vertex of **C**. The algorithm also maintains an ordered list of corners $\mathscr{L}$. Let I be the current first corner of list $\mathscr{L}$. Corner I is the intersection of two supporting lines D_1 and D_2 ($I = D_1 \cap D_2$) corresponding to two contact points p_1 and p_2 that are at present consecutive in the list $\mathscr{C}$. At each step, the algorithm either confirms corner I as being a vertex of **C**, or discovers a new edge lying between p_1 and p_2 on the boundary of **C**. This is achieved by means of at most two probes that are described just below. In the first case we simply report the vertex and delete I from $\mathscr{L}$; in the latter, two new corners are inserted in $\mathscr{L}$. The algorithm halts when $\mathscr{L}$ is empty.

Let us describe precisely the (at most two) probes performed at the current step of the algorithm. The two supporting lines D_1 and D_2 define four wedges R (with p_1 and p_2 on its boundary), S (with p_1 but not p_2 on its boundary), T (with neither p_1 nor p_2 on its boundary), and U (with p_2 but not p_1 on its boundary) (see Figures 2 and 3). Let $\varpi_1 = (p_1, l_1, n_1)$ and $\varpi_2 = (p_2, l_2, n_2)$ be the two probe outcomes whose supporting lines are D_1 and D_2 and let e_1 and e_2 be the edges of **C** containing p_1 and p_2, respectively. The two points p_1 and p_2 are adjacent in the order induced by the set of rays, at this stage of the algorithm. Therefore, from Lemma 1, the region $H_{1,2}$ is known to contain no contact point of the previous probes and, furthermore, a future contact point p is to be inserted between p_1 and p_2 on the boundary of **C** if and only if p lies inside $H_{1,2}$.

The strategy is to exhibit probe paths that will either confirm I as being a vertex of **C** or discover a new edge of **C** between p_1 and p_2. For that purpose, the first probe path μ, issued at the current step is such that:

1. μ aims at I in order to decide whether this point is actually a vertex or not.
2. μ does not intersect the supporting lines D_1 nor D_2, to avoid useless probes with contact points on already discovered edges.
3. The probe segment of μ is guaranteed to lie entirely inside $H_{1,2}$, to ensure that the outcoming contact point will lie between p_1 and p_2 on the boundary of **C**.

Such a probe path μ may be constructed as follows. Let D be a straight line[5] contained in $R \cup T$. Line D passes through I and intersects the segment p_1p_2. We orient D so that p_1 is on the left-hand side of D and p_2 on its right-hand side. The probe path μ is supported by D and its origin o is chosen as follows. The boundary γ of $H_{1,2}$ is a simple closed curve that is the concatenation of the portion of the boundary of **C** between p_1 and p_2, $C_{1,2}$ (unknown at this stage), and of an arc $h_{1,2}$ made of portions of previous probe paths and, possibly, an edge at infinity. Let $o_1, \ldots, o_{2k}$ be the sequence of intersection points between D and $h_{1,2}$, sorted along D. We associate to each intersection point o_i a sign, $+$ if D enters $H_{1,2}$ at point o_i, $-$ otherwise. The origin o of μ is either o_{2k} if o_{2k} has a $+$ sign or the first of two successive intersection points both with $+$ signs. Because γ is a simple closed curve, it follows from Jordan's theorem that such a point exists and, moreover, we are guaranteed that the half-line μ supported by D and starting at o, first encounters **C** at a point p satisfying $op \subset H_{1,2}$. Details can be found in the companion paper [2].

Let $\varpi = (p, l, n)$ be the outcome corresponding to the first probe path μ issued at the current step. The probing ray l is exactly the probe segment op, if o is a point at infinity and, otherwise, the concatenation of op with the infinite portion of the ray l_i ($i = 1$ or 2) passing through o. Lemma 1 implies that p_1, p, p_2 are encountered in that order along the boundary of **C**.

We distinguish four possible cases, depending on whether p belongs to e_1, e_2, both, or none. Notice that, due to Condition 1 above, p belongs to e_i iff p belongs to D_i and $n = n_i$.

Case 1: $p \in e_1$ and $p \in e_2$. In this case $p = I$ and I is confirmed as a vertex of **C**, Due to Condition 1, we are guaranteed that the edges containing p_1 and p_2 are adjacent along the boundary of **C** and that I is their common vertex.

[5] It would be possible to take for D a pseudoline instead of a straight line. This will only affect the complexity of computing the individual probes, not the number of probes.

Case 2: $p \notin e_1$ and $p \notin e_2$. The supporting line $D(\varpi)$ of the probe outcome is distinct from D_1 and D_2. Because p is guaranteed to belong to portion $C_{1,2}$ of the boundary of **C** and because, up to this point, $C_{1,2}$ contains no contact point, a new edge has been discovered.

Case 3: $p \in e_1$ and $p \notin e_2$. In this case $p = I$ but is not a vertex of **C**. Thus probe μ does not confirm I as a vertex of **C** and discovers no new edge. In that case the algorithm issues another probe that is guaranteed to discover a new edge. Let Π_1 be the half-plane on the right-hand side of D_1, when oriented as described above. We distinguish two subcases according to whether p_2 belongs to Π_1 or not. In both subcases, we exhibit a new probe path μ' that is guaranteed to discover a new edge of the boundary of **C** between p_1 and p_2. Path μ' will be supported by a straight line D' passing through I and contained in $S \cup U$.

Subcase 3.1: $p_2 \in \Pi_1$. The situation is depicted in Figure 2. In this case D' is oriented from S to U. Let μ' be the half-line supported by D' and starting at I. The contact point probed by μ' is p'. The corresponding ray l' is the concatenation of Ip' and l. As in Case 2 the new probe necessarily discovers a new edge of **C** (between p_1 and p_2).

Subcase 3.2: $p_2 \notin \Pi_1$. The situation is depicted in Figure 3. We now orient D' from U to S. The origin o' of the new probe path μ' is defined in a way similar to the origin o of μ. This ensures that the new probe necessarily discovers a new edge of **C** (between p_1 and p_2).

Case 4: $p \notin e_1$ and $p \in e_2$. This case is analogous to the previous one. The indices 1 and 2 have simply to be exchanged as well as the wedges U and S.

2.3. Analysis of the Algorithm. Let us count the total number of probes which have been performed. Each step of the probing algorithm either confirms a corner of list $\mathscr{L}$ as a vertex of **C** by means of one probe and this corner will never be probed again, or discovers a new edge by means of at most two probes. Thus to determine the exact shape of **C**, the algorithm issues at most one probe per vertex

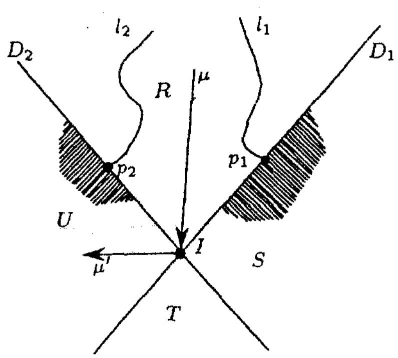

Fig. 2. Case 3.1.

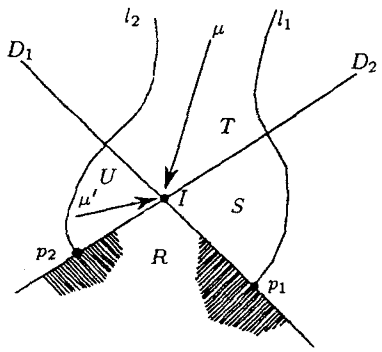

Fig. 3. Case 3.2.

and two probes per edge, except for the first three edges that are discovered in the initialization step by means of only one probe each. This proves the following theorem:

THEOREM 1. $3n - 3$ *probes are sufficient to determine the exact shape of a simple polygon with n noncollinear edges.*

It is proved in [2] that, under our probe model, this bound is also a *lower bound* and that every probe algorithm that determines the shape of a polygon with n edges makes at least $3n - 3$ probes in the worst case. Thus our probing strategy is optimal with respect to the number of probes.

We simply give here an outline of the proof. Remember that, in our model, a probe tangent to an edge misses that edge. Thus polygon **C** is not completely explored as long as we do not have a contact point on each edge and a contact point at each vertex. Indeed, otherwise we might have missed an edge of **C** (of potentially arbitrary small size). Thus a trivial lower bound on the number of probes needed to discover an n-sided polygon is $2n$. Our algorithm meets this bound for convex polygons. Let us consider the case of a nonconvex polygon **C**. Let S be a probing strategy that tries to discover the exact shape of **C** by means of a sequence of probes. Suppose that $i - 1$ probes have already been performed. At this stage, some edges of **C** have been discovered. The rays associated with the probes induce an ordering of the discovered edges (the same as the one on **C**). As in Section 2.2, the intersection I between the supporting lines D_1 and D_2 of two successive contact points is called a corner and is a potential vertex of **C**. Either this corner is a vertex of **C** or some new edge has to be discovered between edge e_1 and edge e_2. Sooner or later, S will have to issue a probe aiming at I in order to decide whether this corner is an actual vertex of **C** or not. When this probe answers with a point that coincides with I but belongs to only one of the edges e_1 or e_2, no new edge has been discovered and no vertex has been confirmed. For the lower bound, we construct a polygon **C** where this adverse situation is encountered $n - 3$ times. This is done by induction on the number of corners which are actual vertices of **C**.

The above strategy guarantees that a minimum number of probes are performed. In order to evaluate the actual complexity of the algorithm, it remains to analyze the complexity of determining each new probe path. It is shown in [2] that the probes can be constructed in such a way that each one can be determined in $O(\log n)$ time. Thus the algorithm has overall $O(n \log n)$ time complexity and requires $O(n)$ storage.

2.4. Probing a Polygonal Room from a Given Point Within the Room. The probing strategy developed above can also be used if we want to find out the exact shape of a polygonal room by probing from the inside of the room. This is possible as soon as a point s inside the room is known: this point may be, for example, the initial position of the probing device.

In this case, each contact point may be associated with a ray joining this contact point to the point s without intersecting the exterior of the room. It can be easily proved that the set of rays induces a total order on the set of contact points that corresponds to the order of these points on the boundary of the room and that Lemma 1 holds.

The initialization step of the algorithm performs three probes issued from this point s: the first two probes are issued from point s along two opposite directions. Let $I = D_1 \cap D_2$ (possibly at infinity) be the corner formed by the supporting lines of the two corresponding probe outcomes. The third probe is issued from the point s along the straight line passing through the points s and I and directed from I to s. The three corresponding contact points p_1, p_2, p_3 belong to three distinct edges of the polygonal room. Then a strict application of the probing strategy described above provides a complete description of the polygonal shape of the room in clockwise order.

3. Probing Several Polygons. In this section the probing strategy developed in Section 2 is extended to apply to the case where several polygons have to be simultaneously explored. More precisely, we assume that the probing device has to compute the shape of k' polygons $C_1, \ldots, C_{k'}$ among a scene of k polygons $(1 \leq k' \leq k)$. Let n denote the total number of edges in the scene. The numbers k and n are unknown and will remain unknown, except in the case $k' = k$.

As before, some mild restrictions on the statements of the problem are assumed in order to ensure that the probing problem is solvable within a finite number of steps. Namely:

1. The oriented supporting lines of the n edges in the scene are all distinct.
2. A target point t_i is given within each polygon to be explored $(t_i \in C_i$ $(i = 1, \ldots, k'))$.

Under these conditions, we prove below that $3n - 3 + k$ probes are sufficient to compute the exact shapes of the k' polygons. Unfortunately, these probes are harder to compute than those of Section 2 and our algorithm requires $\Theta(n)$ time per probe and, thus, has overall time complexity $\Theta(n^2)$.

3.1. Description of the Algorithm. Roughly speaking, the present algorithm for probing several polygons uses the divide-and-conquer paradigm in conjunction with the probing strategy described in Section 2. This strategy, valid for the probing of a single polygon, is applied as long as there is no evidence for the presence of several objects among the current set of contact points. When the presence of more than one object becomes manifest, the probing problem is split into two subproblems that are recursively solved.

Before giving the whole algorithm, we describe its main ingredients and introduce the notions of a probing process and of a separator probe.

In the following we call *probing process* a realization of the probing algorithm for a single polygon (in fact, a slight variant to be described below). As explained in Section 2, the current state of a probing process $\mathscr{P}$ is completely determined by the triplet $(\mathscr{C}, \mathscr{L}, \mathscr{H})$, where the current contour, $\mathscr{C}$, is the circular list of contact points sorted according to the order induced by the rays, $\mathscr{L}$ is the corresponding ordered list of corners, and $\mathscr{H}$ is the set of the polygonal chains, $h_{i,i+1}$, made of portions of probe segments and joining pairs, (p_i, p_{i+1}), of successive contact points.

We call a probe whose outcome reveals that the contact points of $\mathscr{C}$ belong to more than one polygonal object the *separator probe*. Such a probe is either a probe whose contact point p is at infinity (if the probe path encounters no polygon) or a probe whose probe segment op intersects the polygonal chains of $\mathscr{H}$ in, at least, one point o' (between o and p). Indeed, as long as no probe segment op intersects the set of chains $\mathscr{H}$, all the contact points of $\mathscr{C}$ belong to the same cell of the subdivision of the plane induced by $\mathscr{H}$ (or equivalently, by the set of the rays). Therefore, we know from [1] that there exists a simple curve passing through all the contact points without intersecting the rays (except at their endpoints); thus there is no evidence that the contact points found so far belong to several polygons.

The algorithm for several polygons will activate several probing processes. Each probing process will be stopped as soon as a separator probe is encountered. As previously mentioned, the probing process is a variant of the basic algorithm of Section 2. The only difference between the variant and the basic algorithm is an additional test. Indeed, here we need to detect when a separator probe is encountered and, therefore, each time a probe is issued, before updating the triplet $(\mathscr{C}, \mathscr{L}, \mathscr{H})$, we have to check whether the probe segment op intersects one of the segments of the current set $\mathscr{H}$ or not. This simple variant of the basic algorithm will serve as the first main ingredient of the algorithm for several polygons.

When a probing process $\mathscr{P}$ with current state $(\mathscr{C}, \mathscr{L}, \mathscr{H})$ encounters a separator probe, it is stopped and replaced by two secondary processes $\mathscr{P}'$ and $\mathscr{P}''$ with current states $(\mathscr{C}', \mathscr{L}', \mathscr{H}')$ and $(\mathscr{C}'', \mathscr{L}'', \mathscr{H}'')$. These secondary processes will evolve recursively in turn. The construction of $(\mathscr{C}', \mathscr{L}', \mathscr{H}')$ and $(\mathscr{C}'', \mathscr{L}'', \mathscr{H}'')$ from $(\mathscr{C}, \mathscr{L}, \mathscr{H})$ and the separator probe segment op is performed by our second main ingredient, the so-called Procedure SPLIT described below. As is proved in the next section, the current states $(\mathscr{C}', \mathscr{L}', \mathscr{H}')$ and $(\mathscr{C}'', \mathscr{L}'', \mathscr{H}'')$ of $\mathscr{P}'$ and $\mathscr{P}''$ will summarize the whole information (as far as probing is concerned) contained in the current state $(\mathscr{C}, \mathscr{L}, \mathscr{H})$ of process $\mathscr{P}$, and both secondary processes $\mathscr{P}'$ and $\mathscr{P}''$ have no evidence for the presence of several polygons among their respective sets of contact points.

Procedure SPLIT

Input: a probing process $\mathscr{P}$ with current state $(\mathscr{C}, \mathscr{L}, \mathscr{H})$ and a separator probe segment *op*.

Output: two secondary processes $\mathscr{P}'$ and $\mathscr{P}''$ with initial states $(\mathscr{C}', \mathscr{L}', \mathscr{H}')$ and $(\mathscr{C}'', \mathscr{L}'', \mathscr{H}'')$.

1. Find the intersection point o' between the separator probe segment *op* and the segments of the set of chains $\mathscr{H}$ which is closest to o.
2. Split the circular list $\mathscr{C}$ into two circular sublists as follows. Among the two chains of $\mathscr{H}$ containing o, let $h_{i,i+1}$ be the one such that the supporting line D_{op} of *op*, oriented from o to p, comes *into* the region $H_{i,i+1}$ at point o (i.e., o has a $+$ sign according to the sign convention of Section 2.2). Among the two chains of $\mathscr{H}$ containing o', let $h_{j,j+1}$ be the one such that D_{op} comes *out* of the region $H_{j,j+1}$ at point o' (i.e., o' has a $-$ sign). $\mathscr{C}'$ is the sublist of $\mathscr{C}$ going circularly from p_{i+1} *to* p_j while $\mathscr{C}''$ is the sublist of $\mathscr{C}$ going circularly from p_{j+1} to p_i. The list $\mathscr{L}$ is split accordingly (see Figure 4).
3. All the chains from $\mathscr{H}'$ and $\mathscr{H}''$ are inherited without change from the corresponding chains of $\mathscr{H}$ except for the chain $h_{j,i+1}$ of $\mathscr{C}'$ and the chain $h_{i,j+1}$ of $\mathscr{C}''$. The new chain $h_{j,i+1}$ is the concatenation of the portion of the old chain $h_{j,j+1}$ from p_j to o', the segment $o'o$ and the part of the old chain $h_{i,i+1}$ from o to p_{i+1}. Similarly, the new chain $h_{i,j+1}$ is the concatenation of the part of the old chain $h_{i,i+1}$ from p_i to o, the segment oo' and the part of the old chain $h_{j,j+1}$ from o' to p_{j+1}.

We can now give a description of the whole algorithm. During the course of the algorithm, a number of probing processes will be activated. Each probing process is activated with an initial state $(\mathscr{C}_0, \mathscr{L}_0, \mathscr{H}_0)$; the initial state of the first probing process is $(\varnothing, \varnothing, l_\infty)$, where l_∞ is the line at infinity. We distinguish between primary and secondary processes. A *primary process* is a process whose initial lists of contact points $\mathscr{C}_0$ and corners $\mathscr{L}_0$ are empty and thus need to be initialized. At the initialization step, we issue three probes aiming at a given target

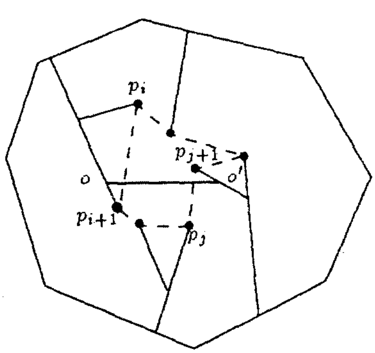

Fig. 4. Illustration of Procedure SPLIT.

(as in the initialization step of the basic algorithm), fill the two lists $\mathscr{C}_0$ and $\mathscr{L}_0$ with the outcomes of these three initial probes, and update $\mathscr{H}_0$.

A *secondary process* is a (already initialized) process which results from splitting a previous process when a separator probe is encountered. A probing process disappears either because its list of corners $\mathscr{L}$ becomes empty, which means that it has completed the exploration of one of the polygons, or because it has been replaced by two secondary processes after a separator probe has been encountered.

At the begining the algorithm activates a primary process with three initial probes aiming at the first target point o_1 as described in Section 2. This primary process and the subsequent secondary processes evolve in turn until all of them have disappeared. We then say that the algorithm has reached a *stable state.*

At such a stage of the algorithm, the exact shape of at least one polygon of the scene has been computed but some of the k' polygons to be explored may have been completely missed, until this point. Assume that, when reaching a stable state, the algorithm has discovered k_1 polygons with altogether n_1 edges. The boundary of these polygons together with the current set of probe segments induce a subdivision of the plane into regions. k_1 of these regions are simply the interiors of the discovered polygons. The others are the regions $H_{i,i+1}$ (called, for short, the H-regions) associated to each pair of contact points (p_i, p_{i+1}) consecutive on the boundary of one of the k_1 polygons.

To ensure discovery of all the k' polygons, the algorithm maintains in a dynamic structure this subdivision of the plane and also a sublist of the given targets o_i which have not yet been located in an explored polygon. Each time a stable probing state is reached, the subdivision of the plane is updated and the algorithm locates in turn each target of the remaining list until it encounters a first target, say t, lying in the interior of an H-region. Then a new primary probing process is activated within this H-region, called the *probing region* of the process. Its initial list $\mathscr{H}_0$ consists of one closed chain,[6] the boundary of the probing region, and the first three probes have their origin on $\mathscr{H}_0$ and aim at target t. This will guarantee discovering at least one new polygon inside the probing region. Such a probing process is very similar to the initial probing process. In fact, both are identical if we consider the line at infinity as the boundary of a special probing region, namely, the whole plane. Due to the usual mechanism, the probes issued by this process have probing segments totally included in the probing region, except possibly for the last probe when it is a separator probe with its contact point outside the probing region.

The whole process is repeated until all the targets have been found to belong to an explored polygon.

3.2. Correctness of the Algorithm. The notion of a probing region, introduced for primary processes, extends in a straightforward way to secondary processes. In all cases the probing region of a probing process is the region of the plane bounded by the initial set of chains $\mathscr{H}_0$ of the process.

---

[6] With possibly an edge at infinity.

As previously noted, all the probe segments issued by a probing process $\mathscr{P}$ are contained in the probing region of $\mathscr{P}$, except possibly for the last probe when it is a separator probe with its contact point outside the probing region. This property guarantees that two probing processes whose probing regions have disjoint interiors are independent. Therefore, the results of Section 2 show that each probing process evolves correctly as long as no separator probe is encountered and we only have to prove the correctness of Procedure SPLIT.

We shall successively prove the three following facts that altogether prove the correctness of Procedure SPLIT:

FACT 1. *Both sublists $\mathscr{C}'$ and $\mathscr{C}''$ are nonempty.*

FACT 2. *The separator probe subdivides the probing region into two subregions with disjoint interiors, one containing the points of $\mathscr{C}'$ and the other the points of $\mathscr{C}''$.*

FACT 3. *The secondary processes $\mathscr{P}' = (\mathscr{C}', \mathscr{L}', \mathscr{H}')$ and $\mathscr{P}'' = (\mathscr{C}'', \mathscr{L}'', \mathscr{H}'')$ together summarize the whole information regarding probing that is contained in the state $(\mathscr{C}, \mathscr{L}, \mathscr{H})$ of $\mathscr{P}$; none of them has any evidence for the presence of several objects among its set of contact points.*

Because no separator probe has been encountered by $\mathscr{P}$ before op, we know that there exists a simple closed curve C joining all the points of $\mathscr{C}$ without intersecting the chains of the set $\mathscr{H}$, except at the points of $\mathscr{C}$. Curve C and the chains of $\mathscr{H}$ together subdivide the probing region of $\mathscr{P}$ into the interior of $\mathscr{C}$ and the regions $H_{i,i+1}$. We denote by γ_i the boundary of the region $H_{i,i+1}$: γ_i is the concatenation of the portion of C, $C_{i,i+1}$, going from the contact point p_i to the contact point p_{i+1}, and of the chain $h_{i,i+1}$ of $\mathscr{H}$ with endpoints p_i and p_{i+1}.

To prove the first and the second facts we consider the intersections between the line D_{op} supporting the separator probe segment op and the set of simple closed curves γ_i. The supporting line D_{op} is oriented from o to p and we assume, for each intersection point between D_{op} and a curve γ_i, the same sign convention as in Section 2: the intersection has a $+$ sign if D_{op} enters $H_{i,i+1}$ at this point and a $-$ sign otherwise. Furthermore, we consider that the intersection points are sorted along D_{op} and, in the following, *first*, *last*, *next*, etc., refer to that order.

PROOF OF FACT 1. We prove that the indexes i and j defined in Step 2 of Procedure SPLIT are distinct, which clearly implies Fact 1. Let us suppose, for a contradiction, that $o' \in h_{i,i+1}$. Due to our conventions, o' is an intersection with a $-$ sign between D_{op} and $h_{i,i+1}$. Thus o is not the last point on the list of intersections between line D_{op} and the chain $h_{i,i+1}$ and, moreover, from the way point o has been chosen on D_{op} (see Section 2.2), this point is the first one of two consecutive intersections between D_{op} and $h_{i,i+1}$, o and o'', both with $+$ signs. Thus point o'' is necessarily between o and o' on D_{op}, which contradicts the fact that o' is the intersection between the probe segment op and the set of chains $\mathscr{H}$ which is closest to o. □

PROOF OF FACT 2. Let l and l' be the rays passing through o and o' respectively and let λ (resp. λ') be the portions of l (resp. l') between o (resp. o') and the common point of l and l' if it exists or, otherwise, infinity. The concatenation of the segment oo' and of λ and λ' is a simple curve, either infinite or closed, whose intersection with the probing region is connected. Let us call such a curve the *separator curve.* This curve subdivides the probing region into exactly two subregions. To complete the proof of Fact 2, we show that one of those subregions contains the points of $\mathscr{C}'$ while the other contains the points of $\mathscr{C}''$. This is done by proving that $C_{i,i+1}$ and $C_{j,j+1}$ intersect the separator curve in an odd number of points while any other $C_{k,k+1}$, for $k \neq i$ and $k \neq j$, intersects the separator curve in an even number of points. Notice first that the intersections between C and the separator curve obviously all belong to oo'. From Jordan's Lemma and the definition of points o and o', the first (along line D_{op}) of these intersections has a $-$ sign and belongs to $C_{i,i+1}$ while the last one has a $+$ sign and belongs to $C_{j,j+1}$. Still from Jordan's lemma, the sequence of signs of the other intersections between C and oo' (if any) is an alternate sequence of $+$ and $-$: $+ - + - \cdots + -$. Let us consider one such intersection with a $+$ sign, belonging, say, to $C_{k,k+1}$. At this point line D_{op} enters region $H_{k,k+1}$ and thus must leave this region later on. From the definition of o and o', D_{op} must leave this $H_{k,k+1}$ through $C_{k,k+1}$, which proves that the subsequent intersection (with a $-$ sign) also belongs to $C_{k,k+1}$. This ends the proof of Fact 2. □

PROOF OF FACT 3. The set of chains $\mathscr{H}'$ and $\mathscr{H}''$ together span the set of chains $\mathscr{H}$, which shows that the two current states $(\mathscr{C}', \mathscr{L}', \mathscr{H}')$ and $(\mathscr{C}'', \mathscr{L}'', \mathscr{H}'')$ together include the whole information (as far as probing is concerned) gathered in the current state $(\mathscr{C}, \mathscr{L}, \mathscr{H})$ of the probing process that disappears. Let us consider curve C' which is the concatenation of the portion $C_{i+1,j}$ of C going counterclockwise from p_{i+1} to p_j and of a curve joining p_j to p_{i+1} obtained by following the chain $h_{j,i+1}$ defined at step 3 of Procedure SPLIT, as closely as possible (Figure 5). Such a curve is a simple closed curve that joins all the contact points of $\mathscr{C}'$ in their order in this sublist and intersects no chain of $\mathscr{H}'$, except at points $\mathscr{C}'$. This proves that the probing process $\mathscr{P}'$, in its current state $(\mathscr{C}', \mathscr{L}', \mathscr{H}')$, has no evidence for the presence of several objects among its contact points. A similar argument holds for the probling process $\mathscr{P}''$ in its current state $(\mathscr{C}'', \mathscr{L}'', \mathscr{H}'')$. □

3.3. Number of Probes. Let us now count the total number of probes performed by the above algorithm. For a polygonal scene including k polygons, at most k separator probes can be encountered. Except for those separator probes, each probe either confirms a corner as being a vertex of one of the polygons or discovers a new edge or guarantees that the next probe will discover a new edge. As a primary probing process starts by aiming at a given target known to belong to a polygon, the first three probes of each primary process are each guaranteed to discover a new edge. At least one such primary process is performed which finally yields the following theorem:

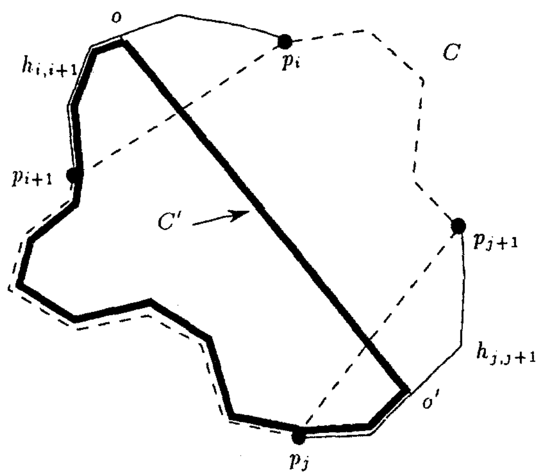

Fig. 5. For the proof of Fact 3.

THEOREM 2. *Given a scene of k polygons including altogether n noncollinear edges, it is possible to determine the exact shape of any subscene of* $k' \leq k$ *polygons by means of at most* $3n - 3 + k$ *probes, provided that one target point is given inside each of these k' polygons.*

It must be noticed that while k' is known from the begining, k and n are in general unknown and will remain so since the algorithm does not determine them.

3.4. Complexity Analysis. A direct consequence of Theorem 2 is that the cardinalities of the sets $\mathscr{C}$, $\mathscr{L}$, and $\mathscr{H}$ are $O(n)$ at any stage of the algorithm. In particular, the set $\mathscr{H}$ of the chains $h_{i,i+1}$ has $O(n)$ edges. This immediately implies that the determination of one probe can be done in $O(n)$ time and thus the determination of all the probes can be done in $O(n^2)$ time. This time bound can be improved to $O(\log n)$ time per probe by using a technique analogous to that of Section 2.3, but this is useless here since the additional test that detects separator probes induces a quadratic complexity.

Indeed, in order to check if the current probe is a separator probe, we have to test if the probe segment op intersects one of the chains of $\mathscr{H}$. This requires examining in turn each segment of the set of chains which takes $\Theta(n_i)$ time for the ith probe. Hence, in total, $\Theta(n^2)$.

Procedure SPLIT is called at most k times. Once all intersection tests have been performed, Procedure SPLIT can be performed in constant time if appropriate pointers link the lists $\mathscr{C}$, $\mathscr{L}$, and $\mathscr{H}$.

Let us now evaluate the complexity of locating the targets t_i, $i = 1, \ldots, k'$, in the successive subdivisions corresponding to the stable states encountered by the probing algorithm. A straightforward induction shows that if k_1 polygons with altogether n_1 edges have been explored, the induced planar subdivision has at most $(3n_1 - 3 + k_1)$ regions. Thus locating a target in the subdivision can trivially be done in $O(n)$ time. Each time a location is queried for a target, either the target is found to belong to one of the explored polygons or a new probing process is activated that will discover a new polygon. Thus $O(k)$ queries are performed, with total cost $O(kn)$. For large values of k, this time bound can be improved to $O(n \log^2 n)$ by using the dynamic structure for maintaining a subdivision described on pp. 135–143 of [6].

The algorithm for probing several polygons is thus dominated by the complexity of the intersection tests which is $\Theta(n^2)$. As mentioned, improving the time complexity of these intersection tests will immediately improve the overall complexity of the method. We left as an open question whether a data structure for storing the set of chain $\mathscr{H}$ can be found that would allow us to perform these tests more efficiently.

3.5. Probing from a Point at Finite Distance. In Section 2.4 we have shown that our basic probing strategy also allows us to compute the shape of a polygonal room as soon as a point inside the room is known. It is easy to see that this is also true in the case of several polygons since the presence of a room does not perturb the evolution of a probing process once this process has been initialized.

We consider the slightly different situation where it is not known in advance whether the objects to be explored are contained in a bounded room or not. Although this problem may appear a bit strange to the reader, it is exactly one of the probing problems that are encountered when probing polyhedra in three-dimensional space. More formally, the problem can be stated as follows: given a scene of polygonal objects, possibly contained in a polygonal room, a point s, lying outside all the objects but inside the room (if any), and k' target points belonging to k' objects in the scene, compute the exact shapes of the k' objects.

We will see that our method can be slightly adapted to solve this problem by means of at most $3n - 2 + k$ probes, where k is the total number of polygons in the scene (including the room, if present) and n is the total number of edges of the scene (including the edges of the room, if present).

Let t_1 be the first target point.

1. The first probe has s as its origin and is directed along the line passing through s and t_1, oriented from t_1 to s. If the contact point is at infinity, no room is present and the usual algorithm described in Section 3.1 can be resumed from the beginning. With respect to the usual algorithm, only one additional probe has been performed. Otherwise, let p_1 be the contact point output by this first probe and let D_1 be the corresponding supporting line.
2. The second probe path is issued along the half-line starting at s and directed toward the target t_1. This probe outputs necessarily a contact point p_2 on the segment Ot_1. Let D_2 be the corresponding supporting line.
3. Let $I = D_1 \cap D_2$ be the corner between the supporting lines D_1 and D_2. The third probe is issued along the half-line starting at s and directed toward I.
   - If this probe path reaches infinity without encountering any obstacle, no room is present and the scene can be probed from infinity. In that case, an

additional probe performed from infinity along the line p_1p_2 and directed toward p_1 (or p_2) is guaranteed to discover a point p on a third edge distinct from the edges containing p_1 and p_2 and the usual probing algorithm can be resumed. (The contact point p_1 (resp. p_2) can now be associated with a ray joining infinity which is the concatenation of the segment op_1 (resp. op_2) with the third probe path.) Once again, only one additional probe has been performed.

- Otherwise, let p_3 be the contact point output by the third probe. The three points p_1, p_2, and p_3 belong to three distinct edges of the scene and form, with the set of chains $\{h_{i,i+1} = p_i o p_{i+1},\ i = 1, 2, 3 \pmod 3\}$, a correct initialization of the first primary probing process. This probing process will be handled in the usual way. Two cases may happen. Either the process will encounter a probing path reaching infinity, at that moment all the contact points may be associated with an infinite ray and the usual probing algorithm can be resumed at that point as in the previous case, or the algorithm will reach a stable state where a number of objects (at least one) and possibly the enclosing polygonal room have been discovered (and thus completely found out). At this stage, any additional primary process which may be necessary can be initialized and further continued in the usual way.

In any case, only the first probe path reaching infinity—this probe proves that no room is present—is an additional probe which has not been counted in the analysis of the basic algorithm. This achieves the proof of the following theorem.

THEOREM 3. *Given*

(i) *a scene of polygonal objects, possibly contained in a polygonal room,*
(ii) *a point s, lying outside all the objects but inside the room (if any), and*
(iii) *k' target points belonging to k' objects in the scene,*

the exact shapes of the k' objects can be computed by means of at most $3n - 2 + k$ probes, where k is the total number of polygons in the scene (including the room, if present) and n is the total number of edges of the scene (including the edges of the room, if present).

REMARK. This algorithm provides the boundaries of the discovered objects in counterclockwise order and the boundary of the polygonal room in clockwise order.

4. Probing Polyhedra in Three-Dimensional Space. The probing algorithm can be extended so as to probe a polyhedron **C** in three-dimensional space. The idea is to discover one edge of **C** at a time by applying another variant of the basic planar algorithm in a plane whose intersection with **C** contains that edge.

This algorithm works under the two following conditions that are the three-dimensional analogous conditions of Conditions 1 and 2 of Section 2:

CONDITION 1. **C** has no collinear edges. Moreover, there is no pair (e,f) where e is an edge of **C** and f is a face of **C** which does not contain e, such that e and f are coplanar (Thus, in particular, **C** will not have coplanar faces).

This condition ensures that no section of **C** through a plane containing e contains edges collinear to e.

CONDITION 2. A target point t belonging to **C** is known.

The probe model is the analog of the probe model used in Sections 2 and 3. When a probe is issued, the probing device responds with the first point where the probe path encounters the object. The probe output includes the contact point, the associated ray, and the normal to the face of the polyhedron passing through this point. The normals are oriented toward the exterior of the object. The sensory device is assumed to be able to detect when the contact point lies on an edge of **C** or is a vertex of **C**, in which cases the normals of all incident faces are reported in the probe output.

4.1. General Outline of the Three-Dimensional Probing Algorithm. We say, as usual, that an edge has been *discovered* when a contact point on this edge has been returned by a probe; furthermore, we say that an edge has been *explored* when its two endpoints have been probed. After an initialization step that discovers a first edge of **C**, the algorithm will consider in turn each discovered edge to find out the vertices of **C** which are its endpoints. Therefore the algorithm maintains the list E of the edges that have been discovered but not yet explored. For each element e in this list, the outcome of the probe that has discovered e (i.e., a contact point on e and the normals to the faces incident to e) is stored. The following pseudocode gives the general outline of the algorithm:

> *Initialization*: First, call Procedure INIT to find a contact point on an edge of **C**. List E is initialized with that edge.
>
> **Loop.** While E is not empty:
>
> 1. Take the first element e of E and call Procedure EDGE(e) to find the vertices of **C** which are the endpoints of e.
> 2. *Remove* e from E and insert in E the edges incident to the endpoints of e and not yet explored (each of these new edges has, as its associated contact point, one of the endpoints of e).

The main ingredients of the three-dimensional probing algorithm are Procedures INIT and EDGE(e), described in Sections 4.3 and 4.4. The aim of Procedure INIT is to issue a contact point on an edge of **C** and the aim of Procedure EDGE(e) is to find the endpoints of the discovered edge e. Both of these procedures choose a plan Π intersecting the object and use a variant of the algorithm described in the previous sections to explore (in general, only partially) the planar section $\Pi(e) \cap \mathbf{C}$. We say, for short, that these procedures *probe in a plane,* which means that all the issued probe paths are included in the same plane.

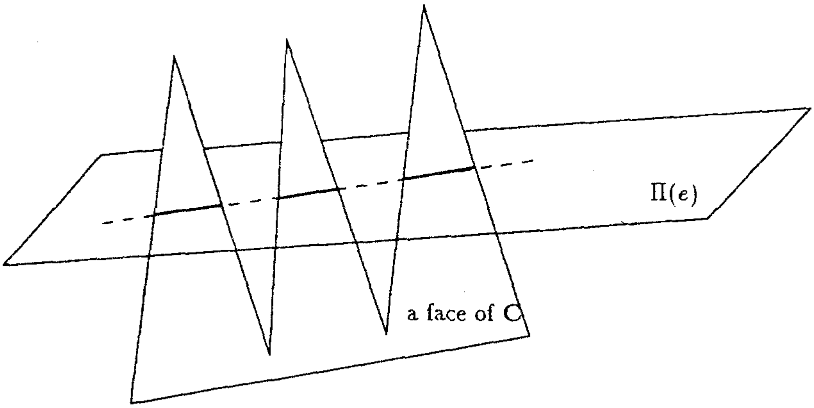

Fig. 6. Collinear edges in a planar section.

The main difficulty encountered at this stage comes from the fact that the planar section $\Pi(e) \cap \mathbf{C}$ does not fulfill Condition 1 of Sections 2 and 3; indeed, it may include collinear edges (see Figure 6). The basic algorithm has no means to understand that two contact points with collinear supporting lines belong to distinct edges unless another contact point has been found on an edge between these two collinear edges. Thus the algorithm is likely to consider two collinear edges as a single one, erroneously too long, and not discover the edges between these collinear edges.

In addition, the current estimate of the polygonal contour (obtained by joining by straight line segments the pairs of consecutive contact points) may be not simple; indeed, the relative interiors of some of its edges may intersect even if their endpoints are confirmed vertices (see Figure 7). This may happen at any stage of a probing process and heavily disturb further evolution of the probing process. To cope with this difficulty, we introduce a variant of the basic algorithm that avoids producing explored edges whose relative interiors intersect. This procedure is described in the next section. It will be used by Procedures INIT and EDGE(e).

4.2. Error Recovery in the Presence of Collinear Edges. Each time both vertices of an edge have been explored, the algorithm checks whether or not the relative interior of this edge intersects some of the edges that have been previously explored. If an intersection is detected, at least one of the two intersecting edges is erroneously long and has to be corrected. This is done by the following procedure.

Let $J = e \cap e'$ be such an intersection. Let $H_{i,i+1}$ (resp. $H_{i',i'+1}$) be the region associated to the contact points p_i and p_{i+1} (resp. $p_{i'}$ and $p_{i'+1}$) preceding and following J on e (resp. e'). We issue a probe aiming at J in one of these regions, say, for example, $H_{i,i+1}$. There are two possible cases:

- If the answered contact point p is not J or coincides with J but does not belong to e (i.e., its normal is distinct from the normal at p_i and p_{i+1}), then e is erroneous:

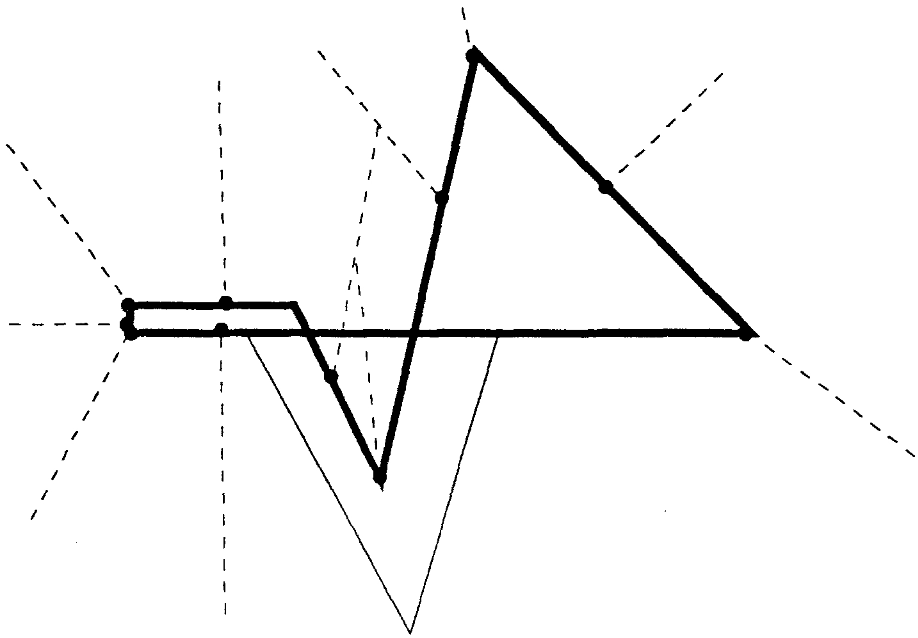

Fig. 7. Intersecting edges with confirmed vertices. —, actual contour; ▬, current estimate of contour; – – –, probe segment; ●, contact point.

p_i and p_{i+1} belong to distinct edges and the contact point p belongs to an edge that has not yet been discovered, lying between p_i and p_{i+1} along the contour.

- Otherwise, the contact point coincides with J and lies on e. Edge e' is necessarily erroneous. A new probe, issued inside $H_{i',i'+1}$ and aiming at J, will necessarily discover a new edge between $p_{i'}$ and $p_{i'+1}$.

In both cases, at least one of the two intersecting edges has been ruled out and, by means of at most two probes, we have discovered a new edge. The probing process can go on as usual.

4.3. Procedure INIT. Procedure INIT chooses a plane Π passing through the target point t and probes in that plane using the probing strategy of Section 3 modified to include the error recovery procedure of the previous section. The first primary probing process is initialized from infinity with point t as its target point. The probing process is stopped as soon as a vertex v of $\Pi \cap \mathbf{C}$ has been confirmed. The edge of $\mathbf{C}$ passing through vertex v of $\Pi \cap \mathbf{C}$ is returned, with the contact point v and the normals to the two incident faces.

Notice that the presence of collinear edges in the planar section $\Pi \cap \mathbf{C}$ does not cause any trouble here since the probing process is not required to explore the whole section but simply to report a vertex.

4.4. Procedure EDGE(e). As soon as e is discovered, the supporting planes of the two faces incident to e are known. Among the four wedges defined by these planes, let R be the wedge which contains **C** in a neighborhood of e and let T be the wedge opposite to R. Procedure EDGE(e) chooses a plane $\Pi(e)$ passing through e and contained in $R \cup T$ and probes in that plane in order to find the vertices of $\Pi(e) \cap \mathbf{C}$ which are the endpoints of e.

The planar section $\Pi(e) \cap \mathbf{C}$ consists of an unknown number of connected components and, moreover, these components may have holes that may themselves include other components. Procedure EDGE(e) has to discover the connected component of $\Pi(e) \cap \mathbf{C}$ which contains e. As this component may be contained in another one, Procedure EDGE(e) uses the probing strategy of Section 3.5. The target point is the contact point $p(e)$ of the probe that discovered edge e. The starting point is obtained by a preliminary probe issued from $p(e)$ along a straight half-line contained in $\Pi(e) \cap T$. Let q be the contact point (possibly at infinity) output by this preliminary probe. Any point s on the segment $p(e)q$ can be taken as a starting point. The probing algorithm is stopped as soon as the component of $\Pi(e) \cap \mathbf{C}$ which contains e has been found.

This component may be erroneous because of the presence of collinear edges in the planar section $\Pi(e) \cap \mathbf{C}$ but the endpoints of edge e are guaranteed to be the actual endpoints of e because, due to Condition 1 of the present section, there is no edge collinear to edge e in the planar section $\Pi(e) \cap \mathbf{C}$.

Notice that, according to the probing strategy of Section 3.5, $p(e)$ will be the contact point of the second probe issued by the first primary probing process. Thus the component of $\Pi(e) \cap \mathbf{C}$ which contains e has surely been explored by the time the probing algorithm reaches its first stable state. Thus no additional primary probing processes will be required, which is fortunate since localization of the target $p(e)$ among erroneous polygonal contours would have been a hazardous undertaking!

4.5. Probing a Scene of Polyhedra. Throughout Sections 4.1–4.4 we have never used the fact that **C** was the unique polyhedron in the scene. Let us suppose that scene **C** consists of k polyhedra satisfying Conditions 1 and 2 of Section 4 and that we want to compute the shapes of a subset of k' polyhedra in the scene located by k' target points. We activate the above algorithm until all the discovered edges have been explored. We have then reached a stable state and computed the shape of some of the polyhedra. The whole algorithm is subsequently rerun, aiming now at a target t', not contained in one of the explored polyhedra (if any). Procedure INIT chooses a plane Π'. Let C' be the intersection of Π' with the already explored polyhedra. If t' is surrounded by a (nonsimply connected) component $C'_{t'}$ of C', Procedure INIT probes inside the hole of $C'_{t'}$ containing t' (i.e., we take this hole as the probing region); otherwise, we use the standard procedure described in Section 4.3. Procedure EDGE(e) is then applied as usual. This procedure is iteratively applied until all the targets have been located inside one of the discovered polyhedra.

4.6. Complexity of the Algorithm. Let us count the number of probes performed by the algorithm. Suppose first that the scene consists of a unique polyhedron **C** with n faces and m edges. Each section of the object has at most n edges and $n/3$ connected components. Indeed, if a section contains zero- or one-dimensional parts (i.e., a vertex or an edge of **C** with all their incident faces on the same side of the cutting plane), our probes will miss them (these are tangent probes); thus any connected component of a cross-section of **C** has at least three edges. Therefore, from Theorems 2 and 3, Procedure INIT and Procedure EDGE perform at most respectively $\frac{10}{3}n - 3$ and $\frac{10}{3}n - 2$ probes. Procedure EDGE is called m times. Therefore, the total number of probes performed by the algorithm is at most $\frac{10}{3}n(m + 1) - 2m - 3$. If the scene consists of k polyhedra with n faces and m edges in total, Procedure INIT is activated at most k times. The total number of probes performed by the algorithm is, in that case, at most $\frac{10}{3}n(m + k) - 2m - 3k$. We sum up our results in the following theorem:

THEOREM 4. *Let S be a scene of k polyhedra with m noncollinear edges, n noncoplanar faces, and such that no edge is contained in the supporting plane of a nonincident face. We can determine, by means of at most* $\frac{10}{3}n(m + k) - 2m - 3k$ *probes, the exact shape of any subscene of k′ polyhedra of S located by k′ target points, one inside each polyhedron.*

5. Concluding Remarks. In this section we discuss our results and present some related open questions. Other open problems on geometric probing can be found in a recent paper [8].

1. The probing algorithm developed by Cole and Yap for convex objects assumes a simple finger probe model whose outcome consists only of the coordinates of a point on the boundary of the object but contains no information on the direction of the normal at that point. We have introduced a new probe model that includes the normals at the contact points.

This seems to be an essential feature for probing nonconvex objects. Indeed, without additional hypothesis, the problem of finding the exact shape of nonconvex polygons with a finite number of finger probes has no solution. Even if collinear points are found, we cannot guarantee that they belong to the same edge of **C**; thus an edge can never be confirmed as an edge of **C**. Nevertheless, we have shown in [2] that, when no information on the normal directions is available, a variant of our method will almost surely output the exact shape of the object, provided that, in addition to the two conditions stated in Section 2, the following third condition is fulfilled:

CONDITION 3. If the intersection point of the supporting lines D_i and D_j of any pair of edges e_i and e_j of **C** belongs to **C**, then it belongs to e_i or e_j.

More precisely, we have the following theorem:

THEOREM 5. *Provided that Conditions* 1–3 *are fulfilled, the above procedure discovers with at most* $8n - 4$ *finger probes a polygon which almost surely is identical to* **C**.

The method obviously extends to the case of several planar objects. It also extends to the case of polyhedra provided that Condition 3 is replaced by the (analogous) following condition: the intersection of the supporting planes of any two distincct faces intersects **C** only finitely many times (in which case we can always slightly rotate the cutting plane so that, in each planar section, Condition 3 is satisfied).

2. In this paper we have mainly tried to optimize the number of probes and have ignored, in our complexity analysis, the cost of moving the probing device from one point to another. Our strategy is not good, in general, for this task and we can exhibit situations, even in the simplest case of one single polygon, where the probing device will be moved along $\Omega(n^2)$ (noncollinear) straight line segments. On the other hand, a probing device that adopts the strategy of moving toward the target until it reaches the object and then follows the boundary of the object, will perform an infinite number of probes to ensure that no edge is missed, but the trajectory followed by the device is clearly the shortest possible one. Between these two extreme situations, there is surely room for interesting compromises. For example, how many probes are necessary and sufficient to determine the exact shape of a planar object using only $O(n)$ turns?

3. Theorem 4 gives an upper bound on the number of probes in the three-dimensional case that is quadratic. Is there also a quadratic lower bound?

4. Lastly, we recall an open question already mentioned at the end of Section 3.4: does a suitable data structure exist that allows us to compute efficiently the probes in the case of several polygons?

Acknowledgments. Jean-Pierre Merlet is gratefully acknowledged for pointing out a mistake in a previous version of this paper and for suggesting to us to look at the problem in three-dimensional space. He is also acknowledged for supplying us with his interactive drawing preparation system draw. We also thank Michel Pocchiola for helpful comments.

References

[1] Alevizos P. D., Boissonnat J. D., Yvinec M., Nonconvex contour reconstruction, *J. Symbolic Comput.* **10** (1990), 225–252.

[2] Alevizos P. D., Boissonnat J. D., Yvinec M., Probing nonconvex polygons, *Proc. IEEE Int. Conf. on Robotics and Automation*, Phoenix, AZ, May 1989.

[3] Bernstein H. J., Determining the shape of a convex n-sided polygon by using $2n + k$ tactile probes, *Inform. Process. Lett.* **22** (1986), 255–260.

[4] Cole R., Yap C., Shape from probing, *J. Algorithms* **8** (1987), 19–38.

[5] Dobkin D., Edelsbrunner H., Yap C. K., Probing convex polytopes, *Proc. ACM Symp. on Theory of Computing*, 1986, pp. 424–432.

[6] Mehlhorn K., *Data Structures and Algorithms III, Multi-dimensional Searching and Computational Geometry*, Springer-Verlag, Berlin, 1984.

[7] Skiena S. S., Geometric Probing, Ph.D. Thesis, Tech. Report No. UIUCDCS-R-88-1425, Department of Computer Science, University of Illinois at Urbana Champaign, IL, April 1988.

[8] Skiena S. S., Problems in geometric probing, *Algorithmica* **4** (1989), 123–127.