Discrete Comput Geom 3:123-136 (1988)

Discrete & Computational Geometry

Separating Two Simple Polygons by a Sequence of Translations*

R. Pollack,[1] M. Sharir,[1,2] and S. Sifrony[2]

[1] Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012, USA

[2] School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel

Abstract. Let P and Q be two disjoint simple polygons having m and n sides, respectively. We present an algorithm which determines whether Q can be moved by a sequence of translations to a position sufficiently far from P without colliding with P, and which produces such a motion if it exists. Our algorithm runs in time $O(mn\alpha(mn) \log m \log n)$ where $\alpha(k)$ is the extremely slowly growing inverse Ackermann's function. Since in the worst case $\Omega(mn)$ translations may be necessary to separate Q from P, our algorithm is close to optimal.

1. Introduction

In this paper we develop an algorithm for the problem stated in the abstract. That is, for a given pair of disjoint simple polygons P and Q having m and n sides respectively, determine whether Q can be moved by a sequence of translations to a position sufficiently far from P without colliding with P, and produce such a motion if it exists. This problem generalizes previous research on translational separability of planar objects (see Toussaint [22], for a survey of this research). In most of the previous work on planar separability the goal was to separate the given objects by translating them one at time in some single fixed

* Work on this paper by the first author has been supported by National Science Foundation Grant No. DMS-8501947. Work on this paper by the second author has been supported by Office of Naval Research Grant No. N00014-82-K-0381, National Science Foundation Grant No. NSP-DCR-83-20085, and by grants from the Digital Equipment Corporation, and the IBM Corporation. Work by the second and third authors has also been supported by a grant from the joint Ramot-Israeli Ministry of Industry Foundation. Part of the work on this paper has been carried out at the Workshop on Movable Separability of Sets at the Bellairs Research Institute of McGill University, Barbados, February 1986.

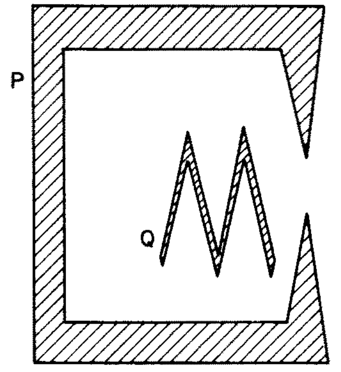


Fig. 1.1. An instance of the polygon separability problem.

direction. In the case of two simple polygons P and Q as above, Toussaint [22] has given an $O(m+n)$ algorithm for determining whether P and Q can be separated by a single translation of one of them (see also Sack and Toussaint [18]).

In this paper we study the problem of separating Q from P under an arbitrary sequence of translations (see Fig. 1.1 for an illustration of this problem). This version of the problem is of intermediate complexity between the simple single-translation separability problems mentioned above and the more difficult problem of separating Q from P by any collision-free motion, involving both translations and rotations. In fact, the problem that we study is a special instance of the motion-planning problem which seeks a purely translational collision-free motion of a polygonal object (Q) amidst a collection of polygonal obstacles. Our case is special because the obstacles consist of a single polygon P. Also, the destination position of Q is fixed (some placement sufficiently far from P); however, our techniques can also handle the case of an arbitrary destination of Q.

Several related motion-planning problems have been recently studied. If the moving object Q is a *convex* polygon (having m sides) and P is an arbitrary collection of polygonal obstacles (having n sides altogether) then one can play a purely translational motion of Q between any two placements (when such a motion exists) in time $O(mn \log mn)$ (see Leven and Sharir [15], Kedem and Sharir [11], Kedem *et al.* [13], Chew and Drysdale [5], and Fortune [7]). In fact, within this time bound one can calculate a discrete representation of the entire space of free placements of Q (all having the same given orientation).

Another related problem is the polygon containment problem, in which, given two polygons P and Q, we wish to determine whether Q can be transformed into a polygon which is entirely contained inside P. Chazelle [3] has shown that if P is convex and only translations of Q are allowed, then the existence of such a placement can be determined in $O(m+n)$ time. Related work on polygon containment by translations is also found in Guibas *et al.* [9], in Fortune [7], and in Edelsbrunner and Welzl [6].

The problems become considerably more difficult when the motion of Q may also involve rotations. Chazelle [3] presents an $O(m^3n^3(m+n)\log(m+n))$ naive algorithm for the general polygon containment problem. If Q is a line segment and P is an aribtrary polygonal region, then the general motion-planning problem for Q can be solved in $O(n^2 \log n)$ time (see Leven and Sharir [14] and Sifrony and Sharir [19]). If Q is a convex polygon (and again P is an arbitrary polygonal region) then the motion of Q can be planned in time $O(mn\lambda_6(mn) \log mn)$ [16], [12], where $\lambda_s(k)$ is an almost linear function of k yielding the maximal number of connected graph portions which compose the lower envelope of k continuous functions, each pair of which intersect in at most s points. Chazelle [3] has shown that if P is also convex, then the existence of a (translated and rotated) copy of Q inside P can be determined in $O(mn^2)$ time. Another recent related work by Yap [23] involves planning the passage of an arbitrary simple polygon P through a "door" (an interval opening in some infinite line obstacle l). Such a motion (which can also be viewed as the separation of P from another polygonal arc Q, chosen so that it overlaps l in a sufficiently large interval, and its endpoints are the endpoints of the door) can be planned in time $O(n^2)$.

As stated above, in this paper we investigate the problem of separating Q from P by a purely translational motion. We develop an algorithm which runs in time $O(mn\alpha(mn) \log m \log n)$, where $\alpha(k)$ is the inverse Ackermann's function. We also exhibit an example in which Q may require $\Omega(mn)$ translations to be separated from P, showing that in the worst case our algorithm is close to being optimal.

We also show that the space F of all free placements of Q (whose combinatorial complexity is always at most $O(m^2n^2)$) can have $\Omega(m^2n^2)$ connected components in the worst case. Thus our algorithm is superior to any motion-planning algorithm that has to calculate the entire space F.

Our algorithm also has the following properties:

(1) Given a final desired separated position of Q, the algorithm can produce the *shortest* separating translational motion of Q from its given position to that destination.
(2) Given an integer k, the algorithm can determine whether Q can be separated from P using at most k translations, and, if so, produce such a "k-separating motion."
(3) The algorithm can be generalized to an algorithm for planning collision-free translational motion of Q between any two free placements.

2. The Algorithm

Our algorithm is based on the following well-known observation (see Lozano Perez and Wesley [17]). Fix a reference point Z in Q, and assume without loss of generality that at the given placement of Q, Z lies at the origin. Define

$$K = P - Q = \{x - y \colon x \in P, y \in Q\},$$

where $x - y$ denotes vector difference (K is known as the Minkowski or vector difference of P and Q). Clearly, a placement of Q (with the same given orientation) intersects P if and only if the reference point Z at this placement lies in

K. Thus the space F of free placements of Q (that is, placements in which P and Q are disjoint) is conveniently represented as $F = K^c$ in the sense that each $x \in F$ corresponds to the free placement of Q in which Z coincides with x.

Thus our goal is reduced to that of analyzing K^c. More specifically, we wish to determine whether the origin (i.e., the given placement of Z) and the point at infinity lie in the same connected component of K^c, and, if so, calculate a (polygonal) path π between these two points which lies entirely within K^c. Note that each straight segment in π corresponds to a single translation of Q, so that the number of segments in π is the number of translations in which Q can be separated from P; also the length of π between O and some point sufficiently far from P is equal to the total translational distance in which Q is moved during this motion.

The approach that we take thus aims to calculate the unbounded connected component C_∞ of K^c. We first establish some properties of K and of C_∞.

Lemma 2.1.

(a) *K is a polygonal region having at most $O(m^2n^2)$ corners.*
(b) *C_∞^c is a simple polygon.*

Proof. (a) We repeat here well-known arguments (see, for example, Guibas *et al.* [9]). Clearly, the boundary of K must consist of vector differences of pairs of points lying, respectively, on the boundaries of P and of Q. Hence $P - Q$ is a polygonal region, each of whose edges has the form $p - q$, where either p is an edge of P and q is a vertex of Q or p is a vertex of P and q is an edge of Q. Since there are at most $2mn$ such differences, it follows that the boundary of $P - Q$ is contained in the union of these $2mn$ segments. Moreover, each corner of $P - Q$ must be either the difference of a vertex of P and a vertex of Q, or a point of intersection of two of the above segments. Since there are plainly at most mn corners of the first kind and $O(m^2n^2)$ corners of the second kind, the claim follows.

(b) It is plain that $P - Q$ is connected, so that the boundary of any connected component of $(P - Q)^c$ must be connected. □

Before continuing we present two examples which help to calibrate the worst-case combinational complexity of $P - Q$ and of C_∞.

Example 1. This example shows that in the worst case $P - Q$ can have $\Omega(m^2n^2)$ connected components (and thus also $\Omega(m^2n^2)$ corners). In this example, as illustrated in Fig. 2.1, Q consists of two "combs" each having m long thin "teeth" so that their "backbones" are perpendicular to one another and their teeth point away from the right angle formed between the backbones. P also consists of a pair of matching aligned combs, each having n teeth, which now point into the right angle formed between the backbones of P. The separation between adjacent teeth of P is taken to be much larger than the separation between adjacent teeth of Q. (Figure 2.1 shows a skeletal representation of P and Q; by slightly thickening these skeletons we can turn them both into simple polygons.) It is clear that if the length of the teeth and backbones of P and Q and the separations between

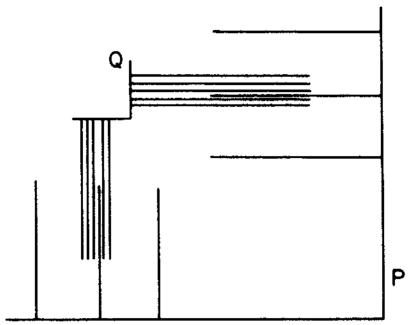


Fig. 2.1. Example 1.

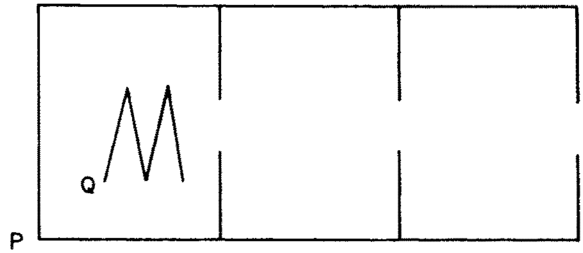
the teeth of P and between the teeth of Q are appropriately chosen, the space $P-Q$ of free positions of Q will contain $\Omega(m^2n^2)$ connected components, each of which (except the unbounded component) is determined by choosing a pair of adjacent vertical teeth of Q and placing them on two sides of a vertical tooth of P, and by similarly choosing a placement of the horizontal teeth of Q amidst those of P.

Example 2. This example shows that in the worst case the boundary of C_∞ can have $\Omega(mn)$ corners. In this example, as illustrated in Fig. 2.2(a), P has a skeletal representation consisting of a sequence of m rectangular "rooms" lying in a row next to each other, such that each pair of adjacent rooms are connected by a small "door" in their common wall, and such that the last (rightmost) room also has a similar door in its right exterior wall. The second polygon Q has a skeletal representation of the shape of a zigzag line consisting of n' segments. The initial placement of Q is in the "innermost" (leftmost) room of P. The dimensions of P and Q can be chosen so that the only way to translate Q out of P is to move it to the right through one door at a time, so that translation of Q through each door must involve n' distinct translations in alternating upward and downward directions, each pushing a different segment of (the skeleton of) Q through the door. Since P has $m = O(m')$ sides and Q has $n = O(n')$ sides, it follows that in this example $\Omega(mn)$ translations may be required to separate Q from P. This example also shows that the unbounded component C_∞ of $(P-Q)^c$ can consist of $\Omega(mn)$ sides, as illustrated in Fig. 2.2(b). We are indebted to Ryan Hayward for suggesting this example.

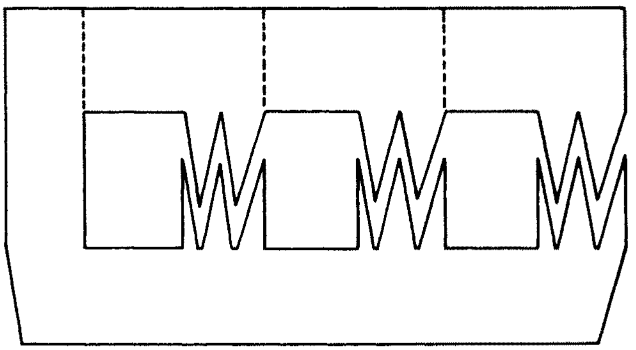
We next show that even though the entire $P-Q$ may have $\Omega(m^2n^2)$ corners, the boundary $\mathrm{bd}(C_\infty)$ of C_∞ contains at most only $O(mn\alpha(mn))$ corners, a bound which is very close to the worst-case lower bound provided by Example 2.

Theorem 2.2. $\mathrm{bd}(C_\infty)$ *has at most* $O(mn\alpha(mn))$ *corners.*

Proof. As observed in the proof of Lemma 2.1(a), $\mathrm{bd}(C_\infty)$ is contained in the union of $2mn$ segments, each of which is either a difference of a side of P and



(a)



(b)

Fig. 2.2. (a) Example 2. (b) $(P-Q)^c = C_\infty$ in Example 2.

a vertex of Q or of a vertex of P and a side of Q. Enumerate these segments as $e_1, e_2, \ldots, e_r$, where $r = 2mn$. It is clear that we can orient each segment e_i so that a sufficiently small neighborhood of e_i lying on its right side is disjoint from C_∞. By Lemma 2.1(b), $\gamma = \text{bd}(C_\infty)$ is a simple closed polygonal curve, and the above observation implies that if we traverse γ in a clockwise direction, then for each segment e_i, evey portion of it that appears along γ is traversed in the direction assigned to e_i; moreover, the (clockwise) order in which these portions are encountered along γ coincides with their order along e_i (we omit details of the proof of these rather simple topological facts).

Consider the circular sequence of straight segments $\gamma_1, \gamma_2, \ldots, \gamma_t$ of which γ is composed (starting at an arbitrary corner of γ and arranged in clockwise order). For each $i \le t$ the segment γ_i is a portion of some e_{u_i}, let U denote the (circular) sequence $u_1, u_2, \ldots, u_t$. For each segment e_a consider all appearances of its index a in U. The preceding arguments imply that there exist two appearances $u_{\text{f}(a)}, u_{\text{l}(a)}$ of a in U, which we denote, respectively, as the *designated first* and the *designated last* appearances of a in U, such that all other appearances of a in U are within its portion $U^{(a)} = (u_{\text{f}(a)}, u_{\text{f}(a)+1}, \ldots, u_{\text{l}(a)-1}, u_{\text{l}(a)})$. (Regarding these notations, recall that U is a circular sequence, so we might have $\text{f}(a) > \text{l}(a)$, in which case $U^{(a)}$ consists of the portion $u_{\text{f}(a)}, \ldots, u_t$ followed by the portion $u_1, \ldots, u_{\text{l}(a)}$; note also that one might have $\text{f}(a) = \text{l}(a)$, in which case a appears in U just once, and $U^{(a)}$ consists of the single element $u_{f(a)}$.) We will regard each $U^{(a)}$ as a linear sequence whose elements are ordered by their circular order along U.

The sequence U has the following properties:

(1) $u_i \in \{1, \ldots, r\}$ for each $i \le t$.
(2) $u_i \ne u_{i+1}$ for each $i \le t$ (where $i+1$ is computed modulo t).
(3) For each pair $a \ne b \in \{1, \ldots, r\}$, there do not exist five indices $p < i < j < k < q$ in $U^{(a)}$ (in the linear order induced on $U^{(a)}$ by U) such that $u_p = u_j = u_q = a$ and $u_i = u_k = b$.

The first two properties are obvious. The third property actually states that for each segment e_a, as portions of it appear along γ in their order along e_a, no other segment e_b can alternate twice between these portions. The proof is topological in nature, and shows that such an alternation would imply that points of C_∞ lie both in the exterior and in the interior of γ, which contradicts the Jordan curve theorem.

More specifically, to establish the third property, assume, to the contrary, that such a, b, and p, i, j, k, q exist. Let us denote by $\gamma_{i,j}$ the portion of γ traversed in clockwise order from γ_i to γ_j, excluding γ_i and γ_j. Without loss of generality, assume that no element of U between u_i and u_k is equal to b, and that no element of U between u_p and u_i or between u_k and u_q is equal to a. We distinguish two cases.

Case (i). The portion e_a^* of e_a between γ_p and γ_q (excluding these two subintervals) does not intersect the portion e_b^* of e_b between its two subintervals γ_i and γ_k (again excluding γ_i and γ_k). Let δ be the closed curve

$$\delta = \gamma_{p,i}\gamma_i e_b^* \gamma_k \gamma_{k,q} e_a^*$$

(see Fig. 2.3). By our assumptions and the fact that γ is simple, it follows that δ is also simple, and is thus a (polygonal) Jordan curve.

We claim that the interior of δ is disjoint from C_∞. Indeed, assume the contrary and let w be a point in the interior of δ which also belongs to C_∞. Then one can find a path π connecting w to the point at infinity, lying entirely within the

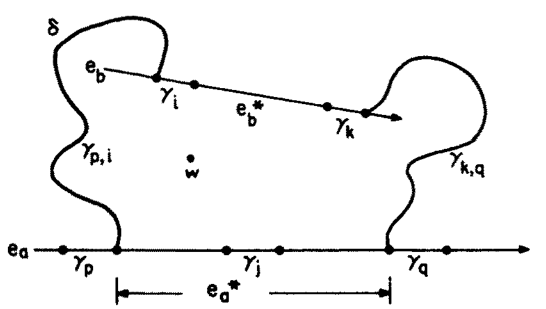


Fig. 2.3. Case (i) in the proof of Theorem 2.2, property (3).

interior of C_∞, and intersecting δ only finitely many times, so that each of these intersections is transversal and takes place at a point in the relative interior of some edge of δ. But each such intersection x is either at an edge of γ, i.e., a point on the boundary of C_∞, or is a point on e_a or on e_b in which case π must contain points (lying sufficiently near x on one side of it) which lie in C_∞^c, contradicting in both cases the assumption that π lies entirely within the interior of C_∞.

Now consider γ_j which is a portion of e_a^*. Since points lying on the left side of γ_j sufficiently near it belong to C_∞, it follows that the interior of δ near γ_j lies on the right side of that edge. But then it is easily checked that both sides of γ_p (also of γ_q) sufficiently near these edges lie in the interior of δ, which is impossible because one of these sides contains points in C_∞. This contradiction completes the argument in Case (i).

Case (ii). e_a^* and e_b^* intersect. Let x denote their point of intersection, and assume without loss of generality that x lies between γ_j and γ_q (see Fig. 2.4). Let δ be the curve

$$\delta = \gamma_{p,i}\gamma_i\bar{e}_b\bar{e}_a,$$

where $\bar{e}_a$ (resp. $\bar{e}_b$) is the portion of e_a (resp. of e_b) between γ_p and x, excluding γ_p (resp. between γ_i and x, excluding γ_i). Arguing as above, it is easy to show that δ is a Jordan curve whose interior is disjoint from C_∞. However, it follows from the structure of δ that either the left side of γ_i or the left side of γ_j (sufficiently near these edges) must lie in the interior of δ, which is impossible because these sides both lie in C_∞. This contradiction completes the proof of property (3).

We next transorm U into another sequence U^* as follows. Replace each index $a \le r$ for which $U^{(a)}$ "wraps around" U (i.e., $\mathrm{f}(a) > \mathrm{l}(a)$), by two distinct symbols a', a'', such that all appearances of a in the subsequence $u_{\mathrm{f}(a)}, \ldots, u_t$ are replaced by a', and all appearances of a in $u_1, \ldots, u_{\mathrm{l}(a)}$ are replaced by a''. The resulting sequence U^* has the same length t as U, and is composed of at most $2r$ symbols. We claim that U^* satisfies the modified properties:

(2*) $u_i^* \ne u_{i+1}^*$ for each $i < t$.

(3*) For each pair of symbols $a \ne b$, there do not exist five indices $i_1 < i_2 < i_3 < i_4 < i_5$ in U^* such that $u_{i_1}^* = u_{i_3}^* = u_{i_5}^* = a$ and $u_{i_2}^* = u_{i_4}^* = b$.

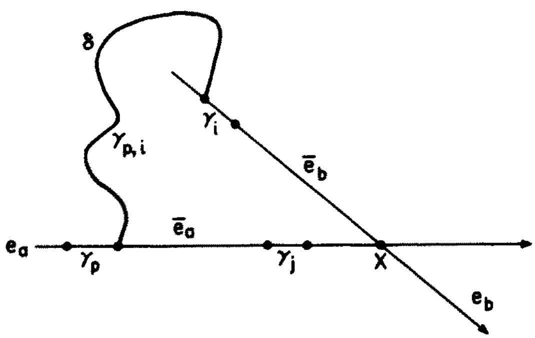


Fig. 2.4. Case (ii) in the proof of Theorem 2.2, property (3).

Indeed, concerning (3*), this property is a direct consequence of (3) if a is one of the original unsplit indices of U. If, say, $a = c'$ for some original index c, then all indices $i_1, \ldots, i_5$ must also belong to the portion of $U^{(c)}$ between the indices $f(c)$ and t, so again the claim follows from property (3) of U. Similar arguments apply if $a = c''$ for some index c.

U^* is thus a $(2r, 3)$-*Davenport-Schinzel sequence* in the terminology of Hart and Sharir [10], and, by the results of that paper, the length of U^* (and of U) is at most $O(2r\alpha(2r)) = O(mn\alpha(mn))$. □

Remark. Using similar arguments to those in the proof of Theorem 2.2, one can obtain the following generalization: let $e_1, \ldots, e_n$ be any (possibly intersecting) n straight segements in the plane. Then the boundary of the unbounded (in fact, of any) component of the complement of the union of these segments consists of at most $O(n\alpha(n))$ segments (which are portions of the segments e_i).

2.1. Efficient Calculation of $\mathrm{bd}(C_\infty)$

We next present an efficient algorithm for the calculation of $\gamma = \mathrm{bd}(C_\infty)$. First obtain a hierarchical decomposition of Q as in Chazelle [2]. Specifically, we first obtain a triangulation T of Q. Then we cut Q along one of the diagonals of T so as to divide it into two subpolygons Q_1, Q_2, each of which contains at least some fixed fraction of the sides of Q, and continue to cut Q_1, Q_2 recursively in the same manner. Chazelle [2] has shown that such a decomposition is always possible; Guibas *et al.* [8] present a linear time algorithm for the calculation of such a decomposition, once a triangulation of Q is given (see Tarjan and Van Wyk [21]).

Our algorithm then applies the following divide-and-conquer approach: Let $Q = Q_1 \cup Q_2$ be the decomposition of Q as described above. Calculate recursively the boundaries γ_1, γ_2 of the unbounded connected component of $(P - Q_1)^c$, $(P - Q_2)^c$, respectively. Then *merge* γ_1, γ_2 to obtain the desired boundary $\gamma = \mathrm{bd}(C_\infty)$ of the unbounded component of $(P - Q)^c$. The merging of γ_1 and γ_2 is performed as follows. Since γ_1 and γ_2 are both simple polygons, we can use the technique of Chazelle and Guibas [4] (see also Guibas *et al.* [8]) to preprocess each of these polygons into a data-structure which supports efficient response to *ray-shooting* queries, where each such query asks for determination of the first point on γ_1 (resp. on γ_2) hit by a ray emerging from some specified point X in a specified direction u. As shown in Chazelle and Guibas [4] and in Guibas *et al.* [8] this shooting problem can be solved in $O(t \log\log t)$ preprocessing time (and $O(t)$ storage) and $O(\log t)$ query time, where t is the number of sides of γ_1 (resp. of γ_2).

Having preprocessed γ_1 and γ_2 in this manner, we next find a starting point X_1 lying on one of these curves and being exterior to the other (e.g., one can take X_1 to be the leftmost vertex among all vertices of γ_1, γ_2). We then begin to trace the desired boundary γ, which is also easily seen to be the boundary of the unbounded connected component of $(\gamma_1 \cup \gamma_2)^c$, from X_1 in a clockwise direction. Suppose we have progressed along γ from X_1 up to some corner X_i of γ. If X_i is a corner of γ_1 or of γ_2, say for definiteness of γ_1, we take the next edge of γ to be traced to be the edge of γ_1 incident to X_i and pointing from it in a clockwise direction along γ_1. Suppose on the other hand that X_i is a point of intersection of an edge e_1 of γ_1 with an edge e_2 of γ_2. Suppose further without loss of generality that our tracing of γ has reached X_i along e_2; then the next edge e of γ to be traced is taken to be e_1 (which has to be traced from X_i in a clockwise direction along γ_1).

In either case we now have an edge e of γ_1 (and of γ) which we want to trace from the point X_i lying on it until the next corner X_{i+1} of γ. To find X_{i+1} we perform a ray-shooting query to find the first point Z on γ_2 hit by the ray emerging from X_i in the direction of (the appropriate portion of) e. If Z lies on e, we put $X_{i+1} := Z$; otherwise we take X_{i+1} to be the appropriate endpoint of e. This tracing process is repeated until we trace the complete boundary γ back to X_1.

The complexity of this merging procedure can be estimated as follows. Let n_i be the number of sides of Q_i for $i = 1, 2$. By Theorem 2.2, γ_i has at most $O(mn_i\alpha(mn_i))$ vertices, so that the preprocessing of γ_1 and γ_2 for the shooting queries is accomplished in overall time

$$O(mn_1\alpha(mn_1)\log\log mn_1) + O(mn_2\alpha(mn_2)\log\log mn_2) = O(mn\alpha(mn)\log\log mn).$$

The starting point X_1 can be calculated in $O(mn\alpha(mn))$ time. The tracing of γ consists of repeated applications of ray-shooting queries, one for each corner of γ. Since γ has at most $O(mn\alpha(mn))$ corners, it follows that the complexity of tracing γ is $O(mn\alpha(mn)\log mn)$.

Let $T(m, n)$ denote the maximal time required to calculate the boundary of the unbounded component of the complement of $P - Q$, where P and Q are simple polygons having m and n sides, respectively. Then we have the following recurrence:

$$T(m, n) \le T(m, n_1) + T(m, n_2) + O(mn\alpha(mn)\log mn)$$

where $n > 3$ and where both n_1 and n_2 are $\ge n/3$ (see Chazelle [2] and Guibas *et al.* [8]). This formula implies

$$T(m, n) = O(nT(m, 3)) + O(mn\alpha(mn)\log mn \log n).$$

To calculate $T(m, 3)$ we make use of the fact that in this case Q is just a triangle, and is therefore convex. Thus calculation of $P - Q$ can be accomplished by the generalized Voronoi diagram approach of Leven and Sharir [15] in time $O(m \log m)$. It follows that

$$T(m, n) = O(mn[\log m + \alpha(mn)\log mn \log n]),$$

and, assuming without loss of generality $m \ge n$, we obtain

$$T(m, n) = O(mn\alpha(mn)\log m \log n).$$

2.2. *Remarks and Open Problems*

(1) The bound $O(mn\alpha(mn))$ on the size of $\mathrm{bd}(C_\infty)$ as given by Theorem 2.2 is not known to be tight in the worst case; the best matching lower bound is that given in Example 2. Moreover, for $n = 3$ or, more generally, for a convex Q it is known that the entire $P - Q$ contains $O(mn)$ corners, of which only $O(m)$ corners are formed by intersection of edges (as in the proof of Lemma 2.1(a); see Kedem and Sharir [11], Kedem *et al.* [13], and Leven and Sharir [15]). Although for all practical purposes the bound $O(mn\alpha(mn))$ can be considered to be the same as $O(mn)$, it is still interesting from a theoretical point of view to improve it, at least in some special case, e.g., when Q can be decomposed as the union of a small number of openly disjoint convex subpolygons. Alternatively, can one show that in the worst case $\mathrm{bd}(C_\infty)$ can indeed consist of $\Omega(mn\alpha(mn))$ corners?

(2) The merging procedure described above deserves some comments. First, it is significant that this procedure does not calculate all intersections between γ_1 and γ_2 (of which there might conceivably be $\Omega(m^2n^2)$). Note also that if γ_1, γ_2, or γ were not connected, then the complexity of our procedure would deteriorate, either because we would have to spend more time in locating a starting point on each component of γ, or because the shooting queries would require more time, because the technique of Chazelle and Guibas [4] applies only to simple polygons. Finally, it is still an open problem whether γ can be calculated in time linear in the number of sides of γ_1, γ_2, and γ.

(3) Theorem 2.2 and the calculation of $\mathrm{bd}(C_\infty)$ can be generalized to apply to any connected component of $(P - Q)^c$. In particular, we can apply them to the connected component C of $(P - Q)^c$ which contains the given position of (the reference point Z on) Q. Thus in time $O(mn\alpha(mn) \log m \log n)$ we can calculate the space of all free placements of Q reachable from its given position by collision-free translational motion.

3. Calculating Motions Separating Q from P

In this section we comment on the actual calculation of the desired translational motion separating Q from P, using the data structures calculated in the previous section. Let $\gamma = \mathrm{bd}(C_\infty)$ be the (simple polygonal) boundary of the unbounded connected component C_∞ of $(P - Q)^c$, and let $t = O(mn\alpha(mn))$ be the number of sides of γ. To determine whether Q can be separated from P by a purely translational motion from a given placement, we simply have to test whether the position O of the reference point on Q at that placement lies in C_∞. This can be trivially done in $O(t)$ time. If O does indeed lie in C_∞, we can calculate an "optimal" translational separating motion of Q in the sense of one of the following two approaches:

(i) *Calculate a Euclidean Shortest Separating Motion.* Let γ^* be the convex hull of γ. We will follow the reasonable convention that Q and P are considered to be fully separated whenever O lies on or outside γ^*. Again we can test whether

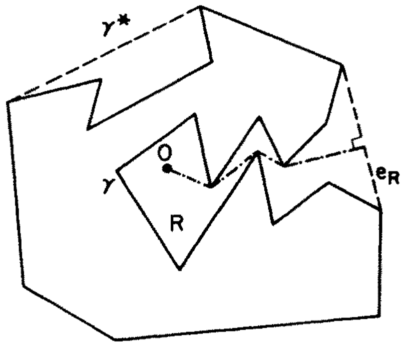


Fig. 3.1. The curves γ, γ^*, and a shortest separating motion of Q from a position within some pocket R.

O lies on or outside γ^* in $O(t)$ time. If so, no separating motion is required. Otherwise O must lie in one of the "pockets" enclosed between γ and γ^*, where each such pocket R is a simple polygonal region bounded by an edge e_R of γ^* (the "lid" of R) not belonging to γ and by a contiguous portion of γ (see Fig. 3.1). In this case our goal is to calculate the shortest path within R from O to a point in e_R. This can be done in $O(t_R \log\log t_R)$ time, where t_R is the number of sides of R, using the shortest path algorithm of Guibas *et al.* [8]. This algorithm calculates the shortest paths from O to all vertices of R, and produces a partitioning of R into triangular regions such that for each of these triangles Δ there exists a vertex v of R such that the terminal segments of the shortest paths from O to all the points in Δ all emerge from v. Hence by examining each triangle in this partitioning which intersect e_R, it is straightforward to calculate the desired shortest path to e_R in $O(t_R \log\log t_R)$ time.

(ii) *Calculate a Separating Motion Consisting of the Smallest Number of Translations.* This is also considered to be a useful criterion for efficiency of the separating motion (see Toussaint [22]). We will say that P and Q are *k-separable* by translations if they can be separated by a sequence of k translations, but not by any sequence of fewer translations; such a sequence of translations will be called a *k-separating motion* (of Q from P). To find such a k-separating motion, we first test, as in (i) above, whether O lies outside γ^*, in which case $k=0$ and no motion is required. If O lies inside γ^*, let R be the pocket containing O (as in (i)) and let e_R be its lid. Our task is then to find a polygonal path within R from O to e_R consisting of the fewest possible number of edges. This problem has been studied recently by Suri [20]. To describe his results, let us partition R into a collection $V_i(e_R)$, $i=1,2,\ldots,$ of polygonal regions defined as follows. $V_1(e_R)$ consists of all points in R directly visible (within R) from some point on e_R. Inductively, $V_{i+1}(e_R)$ consists of all points in R which are visible from some point in $V_i(e_R)$ and are not contained in $\bigcup_{j<i} V_j(e_R)$. Suri shows that this partitioning of R can be calculated in time $O(t_R \log\log t_R)$. Given this partitioning, we find (in additional $O(t_R)$ time) the region $V_k(e_R)$ containing O. Then,

clearly, P and Q are k-separable from one another, and the data structures produced by Suri's algorithm enable us to calculate a k-separating motion of Q in additional $O(t_R)$ time.

In summary, we have shown:

Theorem 3.1. *After calculating γ, one can determine in additional time $O(t_\gamma \log \log t_\gamma)$ (where t_γ is the number of corners of γ) whether Q can be separated from P by translations, and, if so, also calculate such a separating motion having either a minimum length or a minimum number of links.*

Remark. The two approaches just described can be easily modified so that they first perform a preprocessing phase, which depends only on the shape of P and Q and not on their present placements, and then, given specific placements of P and Q, determine quickly whether translational separability of Q from P from these placements is possible, and, if so, also calculate the shortest Euclidean length of such a separating motion, or, alternatively, the smallest number of links in such a motion. Using the techniques of Guibas *et al.* [8] and of Suri [20], such a preprocessing can be done in $O(t \log \log t)$ time and $O(t)$ space, and each actual separability query can be answered in $O(\log t)$ time.

Acknowledgments

We wish to thank János Pach for some helpful discussions on this problem, and for providing us with an alternative proof of some of the main steps in the proof of Theorem 2.2. Thanks are also extended to Godfried Toussaint, and to several of the other participants in the workshop on movable separability of sets, for helpful discussions and for some useful references on this problem.

References

1. B. K. Bhattacharya and G. T. Toussaint, A Linear Algorithm for Determining Translation Separability of Two Simple Polygons, Technical Report SOCS-86.1, McGill University, 1986.
2. B. Chazelle, A theorem on polygon cutting with applications, *Proceedings of the 23th IEEE Symposium on Foundations of Computer Science*, 339–349, 1982.
3. B. Chazelle, The polygon containment problem, in *Advances in Computing Research*, Vol. I (F. P. Preparata, ed.), 1–32, JAI Press, 1983.
4. B. Chazelle and L. Guibas, Visibility and intersection problems in plane geometry, *Proceedings of the ACM Symposium on Computational Geometry*, 135–146, 1985.
5. L. P. Chew and R. L. Drysdale, III, Voronoi diagrams based on convex distance functions, *Proceedings of the ACM Symposium on Computational Geometry*, 235–244, 1985.
6. H. Edelsbrunner and E. Welzl, Private communication, 1985.
7. S. Fortune, A fast algorithm for polygon containment by translation, *Proceedings of the 12th International Colloquium on Automata, Languages and Programming*, 189–198, Lecture Notes in Computer Science, Vol. 194, Springer-Verlag, New York, 1985.
8. L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. E. Tarjan, Linear time algorithms for shortest path and visibility problems inside simple polygons. *Proceedings of the second ACM Symposium on Computational Geometry*, 1–13, 1986.

9. L. Guibas, L. Ramshaw, and J. Stolfi, A kinetic framework for computational geometry, *Proceedings of the 24th IEEE Symposium on Foundations of Computer Science*, 100–111, 1983.
10. S. Hart and M. Sharir, Nonlinearity of Davenport-Schinzel sequences and of generalized path compression schemes, *Combinatorica* **6** (1986), 151–177.
11. K. Kedem, and M. Sharir, An efficient algorithm for planning translational collision-free motion of a convex polygonal object in 2-dimensional space admidst polygonal obstacles, *Proceedings of the ACM Symposium on Computational Geometry*, 75–80, 1985.
12. K. Kedem and M. Sharir, An Efficient Motion Planning Algoritm for a Convex Rigid Polygonal Object in 2-Dimensional Polygonal Space, Technical Report 253, Computer Science Department, Courant Institute, October 1986.
13. 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** (1986), 59–72.
14. D. Leven and M. Sharir, An efficient and simple motion-planning algorithm for a ladder moving in two-dimensional space amidst polygonal barriers, *Proceedings of the ACM Symposium on Computational Geometry*, 221–227, 1985 (also to appear in *J. Algorithms*).
15. D. Leven and M. Sharir, Planning a purely translational motion of a convex object in two-dimensional space using generalized Voronoi diagrams, *Discrete Comput. Geom.* **2** (1987), 9–31.
16. D. Leven and M. Sharir, On the number of critical free contacts of a convex polygonal object moving in two-dimensional polygonal space, *Discrete Comput. Geom.*, to appear.
17. T. Lozano Perez and M. Wesley, An algorithm for planning collision-free paths among polyhedral obstacles, *Comm. ACM* **22** (1979), 560–570.
18. J. R. Sack and G. Toussaint, Separability of pairs of polygons through single translations, manuscript.
19. S. Sifrony and M. Sharir, A new efficient motion-planning algorithm for a rod in two-dimensional polygonal space, *Algorithmica*, to appear.
20. S. Suri, Finding minimum-link paths inside a simple polygon, *Comput. Vision, Graphics Image Process.*, to appear.
21. R. E. Tarjan and C. Van Wyk, An $O(n \log \log n)$ time algorithm for triangulating simple polygons, manuscript, August 1986.
22. G. T. Toussaint, Movable separability of sets, in *Computational Geometry* (G. T. Toussaint, ed.), 335–375, North-Holland, Amsterdam, 1985.
23. C. K. Yap, How to Move a Chair Through a Door, Technical Report 238, Computer Science Department, Courant Institute, August 1986.

Received April 16, 1986, and in revised form November 24, 1986.