On Separating Two Simple Polygons by a Single Translation*

Godfried T. Toussaint
School of Computer Science
McGill University
Montreal, Quebec, Canada

ABSTRACT

Let P and Q be two disjoint simple polygons having n sides each. We present an algorithm which determines whether Q can be moved by a single translation to a position sufficiently far from P, and which produces all such motions if they exist. The algorithm runs in time O(t(n)) where t(n) is the time needed to triangulate an n-sided polygon. Since Tarjan and Van Wyk have recently shown that t(n) = O(n log log n) this improves the previous best result for this problem which was O(n log n) even after triangulation.

1. Introduction

Spurred by developments in spatial planning in robotics, computer graphics, and VLSI layout considerable attention has been devoted recently to the problem of moving polygons in the plane without collisions [1]-[11]. A typical problem in robotics is the FIND-PATH problem [12], where a robot must determine if an object, modeled as a polygon in the plane, can be moved from a starting position to a goal state without collisions occurring between the object being moved and the obstacles. Much work has been done on the problem of hypothesizing channels through free space when the obstacles are convex polygons[13]. For nonconvex objects the problem is bypassed by considering the convex hulls of the objects to be the objects themselves. Thus a crucial aspect of robotics for the geometric modeling needed for spatial reasoning and spatial planning is the representation and recognition of the possible types of movement allowed by different nonconvex shapes [14]. For a survey of movability problems in computational geometry see [15] and for a survey of the relation of computational geometry to robotics see [16].

A robotics problem more closely related to the problem considered in this paper is grasping an object with a robot hand. Ignoring several factors such as forces and friction, and severing the hand from the arm leads to some purely geometrical problems. In particular, if we consider only two-dimensional space and model the "hand" and the "object" as two polygons then an interesting geometrical problem consists of determining for a given "hand"-"object" configuration whether the "hand" is truly grasping the "object," i.e., whether the two polygons are interlocked. Several results along these lines are surveyed in [15]. Interesting simplifications occur when the polygons have additional structure [17].

* This research was supported by NSERC Grant A9293, FCAR Grant EQ-1678, and a Killam Fellowship from the Canada Council.

In this paper we present an efficient algorithm for the problem stated in the abstract. That is, for a given pair of disjoint simple polygons P and Q each having n sides or vertices determine whether Q can be moved by a single translation to a position sufficiently far from P without colliding with P and produce all such motions if they exist. The problem was first considered by Toussaint and Sack [5] who showed that it could be solved in $O(n^2)$ time. Later this result was improved to O(n log n) time [18], [19]. The approach used in [18] and [19] is via *point-location* in *planar subdivisions* [20]. The region outside P but inside the convex hull of P is decomposed into a subdivision such that when a vertex of Q falls in this region its directions of translation can be determined in constant time. However, finding the region in which the query vertex lies takes O(log n) time. This is done for all vertices of Q. The entire procedure is repeated with the roles of P and Q reversed. Finally, the movability of the polygons is determined from the movability of the vertices.

A more difficult problem is that of determining 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. Pollack *et al.* [21] present an algorithm for solving this problem in time $O(n^2 \alpha(n^2)\log^2 n)$ where $\alpha(k)$ is the extremely slowly growing inverse Ackermann's function. Since in the worst case $\Omega(n^2)$ translations may be necessary to separate Q from P, their algorithm is close to optimal.

In this paper we give an algorithm for solving the single-translation problem in time O(t(n)) where t(n) is the time needed to triangulate an n-sided polygon. Since Tarjan and Van Wyk [32] have recently shown that t(n) = O(n log log n) this represents an improvement over the previous best algorithm which required O(n log n) time even after triangulation [19].

2. Geodesic Paths and Relative Convex Hulls

Let $P = (p_1, p_2,..., p_n)$ and $Q = (q_1, q_2,..., q_n)$ be two simple polygons in the plane with non-intersecting interiors. Clearly, the cardinalities of P and Q need not be equal but this assumption simplifies notation. We assume that the polygons are given in standard form, i.e., their vertices, specified in terms of cartesian coordinates, are listed in clockwise order, i.e., the interior always lies to the right of each edge as the polygon is traversed and no three consecutive vertices are collinear. We say that P and Q *are separable under translation* (or more succinctly *separable*) if there exists a *direction* θ such that Q can be translated in direction θ an arbitrary distance without *colliding* with P. By a *direction* we mean an equivalence class of *oriented* parallel lines. In some of the concepts to be defined later we use the notion of direction to mean simply an equivalence class of parallel lines. When this is the case we explicitly use the term *unoriented direction*. Two polygons P and Q *collide* if at some instant in time, during the motion, their interiors intersect, i.e., $\mathrm{int}(P) \cap \mathrm{int}(Q) \neq$ *the null set*.

Given a polygon P and two points $a, b \in P$, the shortest path (or *geodesic* path) between a and b is a polygonal path connecting a and b which lies entirely in P such that the sum of its euclidean edge-lengths is a minimum over all other internal paths. We denote it by $GP(a, b \mid P)$ where the direction is from a to b (see Fig. 1). Geodesic paths find application in many areas such as image processing [22], operations research [23], visibility problems in graphics [24], and robotics. Recently, Chazelle [25] and Lee and Preparata [23] independently discovered the same O(n log n)

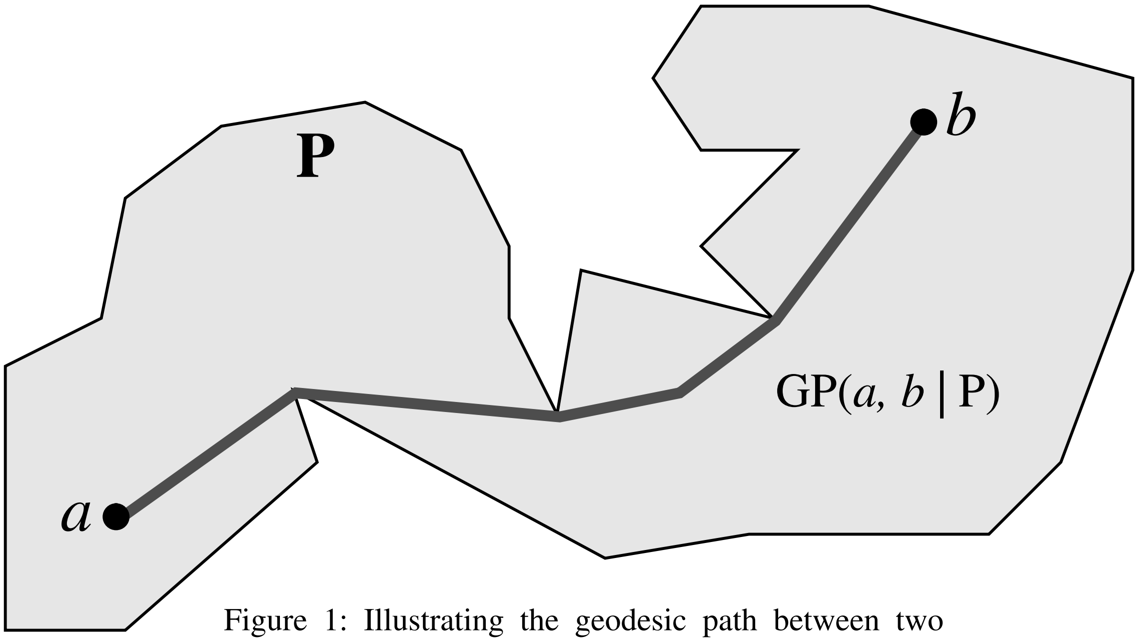

Figure 1: Illustrating the geodesic path between two points a and b in a simple polygon P.

algorithm for computing GP(a, b | P). Both of these algorithms first triangulate P and then find the shortest path in O(n) time. More recently, an algorithm due to ElGindy [26] computes GP(a, b | P) *without* first triangulating P.

Definition: A *polygonal circuit* is a closed polygonal path without self-proper crossings. (This is a slight generalization of the notion of a *simple polygon* to allow some vertices and edges to be used more than once.) Thus it makes sense to speak of its interior and exterior [27]. Accordingly, we also refer to this as a *weakly-simple* polygon.

Definition: The *convex hull* of P relative to Q, denoted by CH(P | Q) is the *shortest polygonal circuit* (or *geodesic circuit*) which contains P and excludes Q; i.e., int(P) $\subseteq$ int(CH(P | Q)) and int(Q) $\subseteq$ ext(CH(P | Q)). Figure 2 illustrates two polygons and CH(P | Q). Figure 3 illustrates a case where the CH(P | Q) is not a simple polygon. We also refer to CH(P | Q) and CH(Q | P) as *relative convex hulls*.

3. Geodesic Circuits and Separability of Polygons

In this section we present the main result of this paper: we show that, given two nonintersecting simple polygons P and Q, the translation separability problem can be reduced to computing the relative convex hulls of P and Q. This result is expressed by Theorem 3.1 below. First we introduce some more notation.

The sides of polygon P, called *edges*, are denoted by $e_j=(p_j, p_{j+1})$ and are directed from p_j to p_{j+1} (indices are modulo n throughout). A *chain* $C_{ij}(P) = (e_i, e_{i+1},..., e_{j-1})$ is a sequence of *edges* on the boundary of P. Similarly, for Q we have $f_j = (q_j, q_{j+1})$ and $C_{ij}(Q) = (f_i, f_{i+1},..., f_{j-1})$. A chain $C_{ij}(P)$ is *monotonic with respect to direction* θ if the projections of the vertices $p_i, p_{i+1},..., p_j$ onto a line

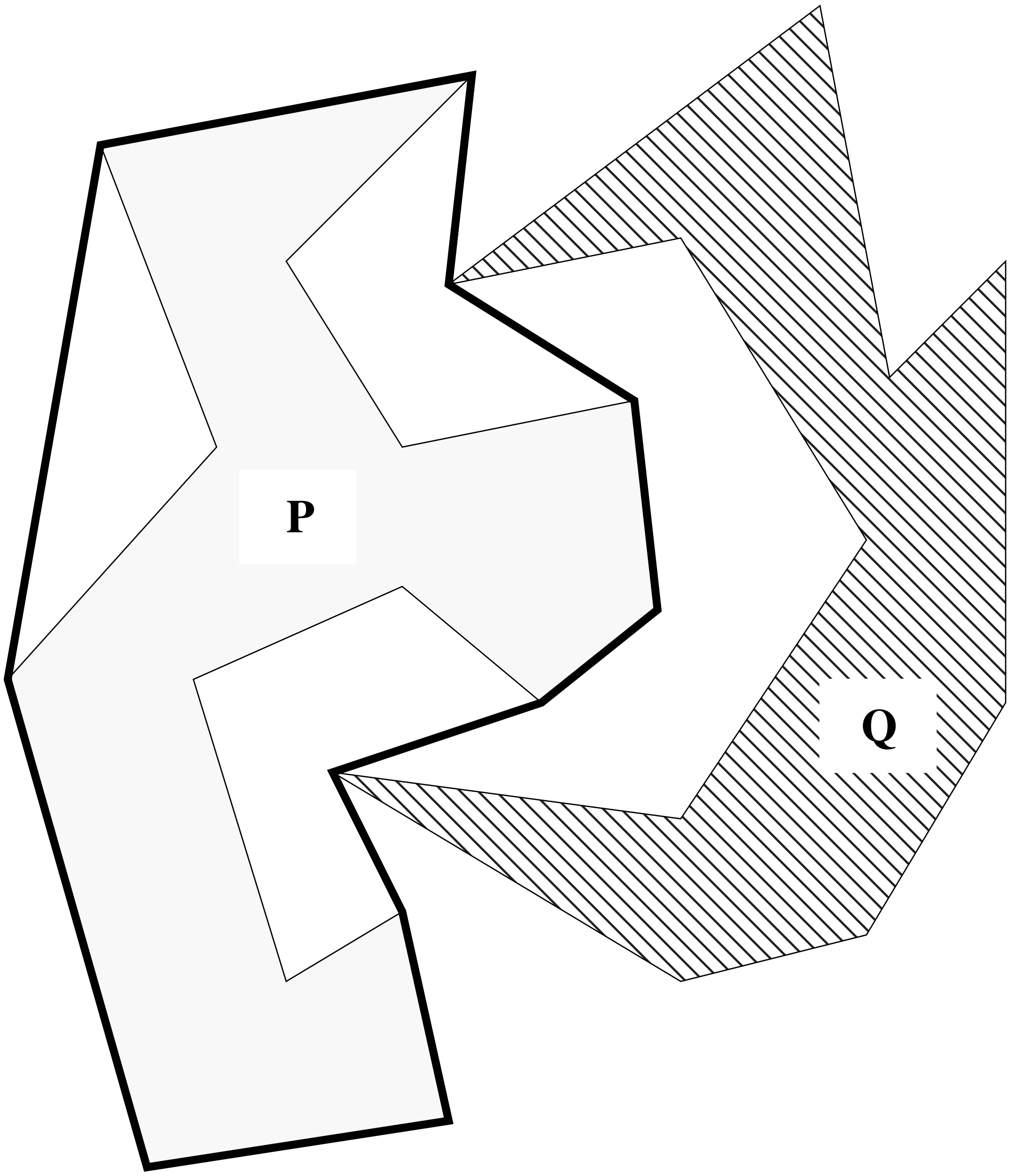

Figure 2: Illustrating the relative convex hull CH(P/Q).

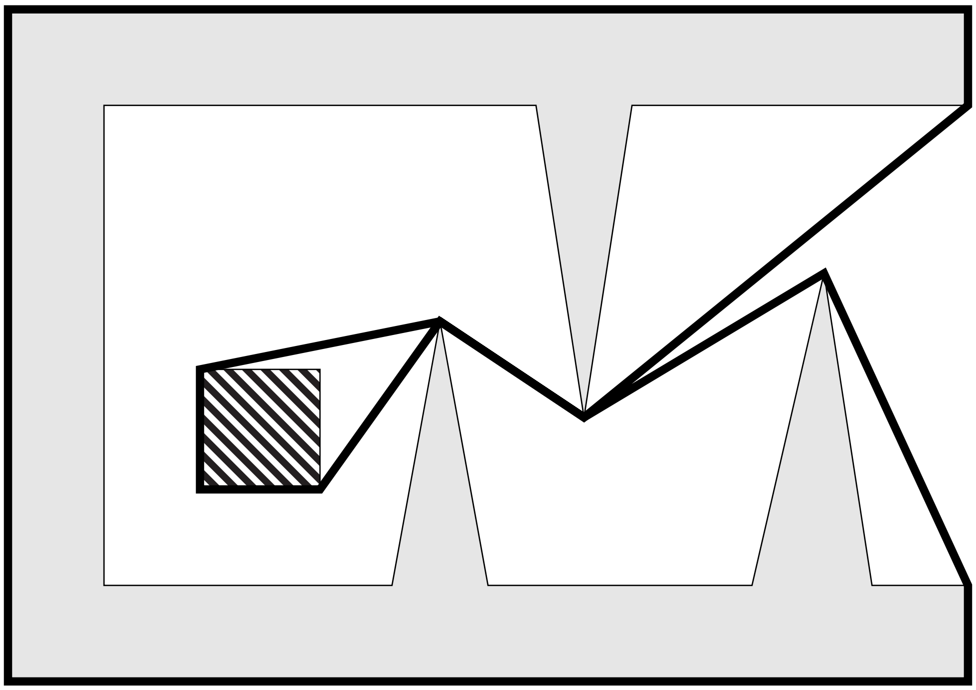

Figure 3: A case when the relative convex hull is a *weakly-simple* polygon.

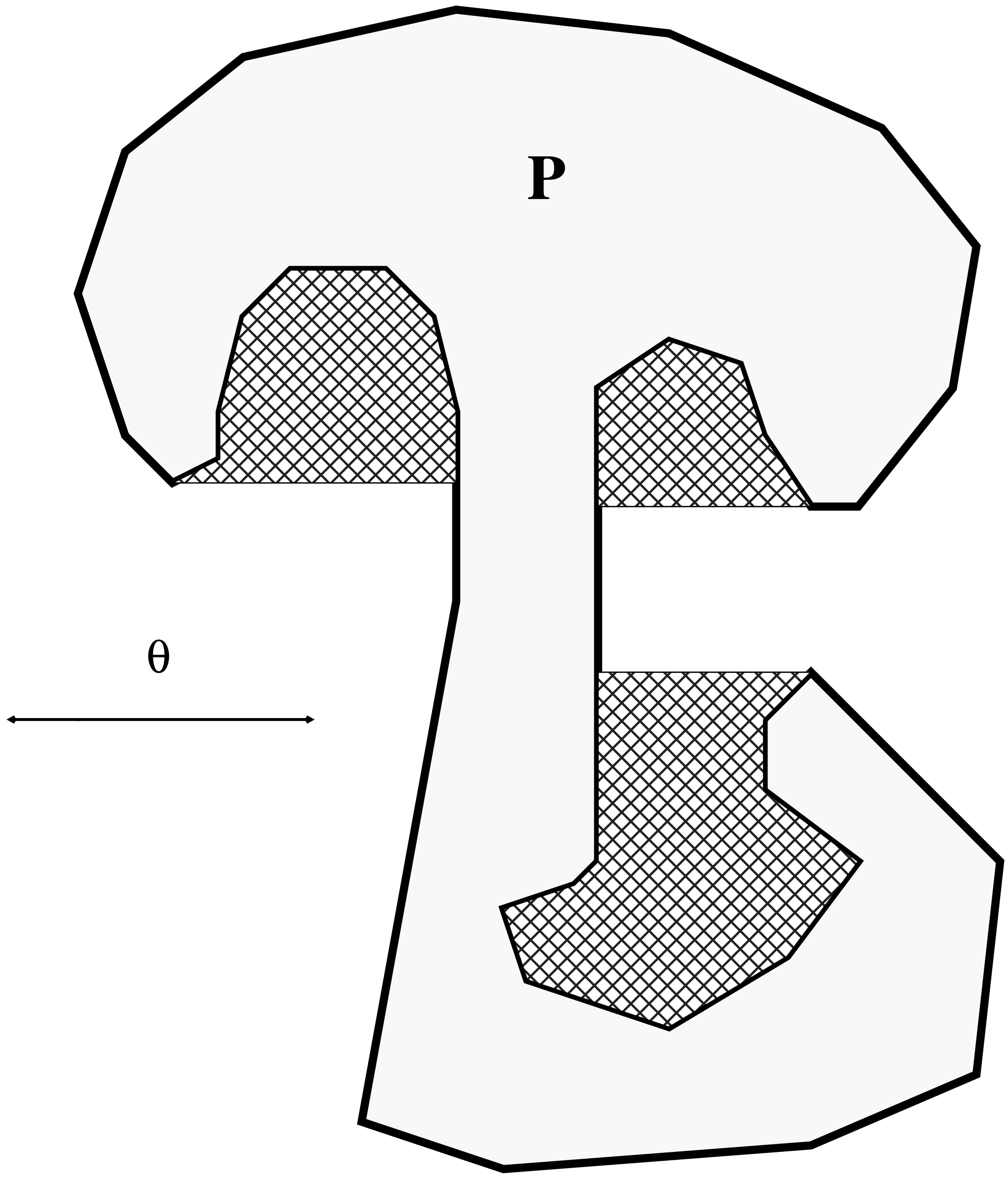

Figure 4: Illustrating the visibility hull of P in direction θ.

$L(\theta)$ in unoriented direction θ are ordered as the vertices in $C_{ij}(P)$. P is a *monotonic polygon* if there exists a line $L(\theta)$ such that the boundary of P can be partitioned into two chains $C_{ij}(P)$ and $C_{ji}(P)$ that are monotonic with respect to θ.

We can now state the main result.

Theorem 3.1: Two disjoint simple polygons P and Q are separable under translation if, and only if, their relative convex hulls are monotonic polygons.

Before proving Theorem 3.1 we need a few lemmas.

Definition: Given a simple polygon P and an unoriented direction θ, the *visibility hull* of P in direction θ, denoted by $VH(P, \theta)$, is the set obtained by taking the union of P with all line segments (a, b) in direction θ such that $a, b \in P$. Note that $VH(P, \theta)$ is monotonic with respect $\theta + 90^o$. The edges on the boundary of $VH(P,\theta)$ which are not edges of P specify a set of "pockets" of $VH(P,\theta)$. See Fig.4 for an illustration of the visibility hull of P and its "pockets."

Lemma 3.1: [17]. Two disjoint monotonic polygons P and Q are separable under translation.

Lemma 3.2: Let P be a polygon monotonic in the unoriented direction θ and let a, b be any two points in P. Then the geodesic path $GP(a, b \mid P)$ is a polygonal chain monotonic with respect to θ.

Proof: Let $GP(a, b \mid P)$ be nonmonotonic with respect to θ. Without loss of generally assume θ to be parallel to the x axis and let a have smaller x coordinate than b. This implies that there exists a pair of consecutive vertices of $GP(a,b \mid P)$, say u, v, such that u occurs before v on a traversal of $GP(a, b \mid P)$ from a to b, and v has smaller x coordinate than u. But since all the vertices of $GP(a, b \mid P)$, other than a and b, coincide with vertices of P, it follows from the Jordan Curve Theorem that the boundary of P contains a chain $C_{ij}(P)$ which is not monotonic with respect to θ. Q.E.D.

Lemma 3.3; [5]. Two disjoint simple polygons P and Q are separable under translation in unoriented direction θ if, and only if,

$$int[VH(P,\theta)] \cap int[VH(Q,\theta)] = \varnothing$$

We are now ready to prove Theorem 1.

Proof: (of Theorem 3.1.) [if part] If $CH(P \mid Q)$ and $CH(Q \mid P)$ are monotonic, then it follows from Lemma 3.1 that they can be separated under translation. Now, by definition $CH(P \mid Q)$ contains P and $CH(Q \mid P)$ contains Q. Therefore, P and Q are separable under translation.

[only if part] We must show that if P and Q are separable then *both* relative hulls are monotonic. Assume therefore that P and Q are separable in unoriented direction θ. Compute the visibility hulls $VH(P,\theta)$ and $VH(Q,\theta)$. From Lemma 3.3 it follows that the interiors of these visibility hulls do not intersect. Now construct the relative convex hulls of the visibility hulls $CH[VH(P,\theta)|VH(Q,\theta)]$ and $CH[VH(Q,\theta) \mid VH(P,\theta)]$. These relative hulls must also be the

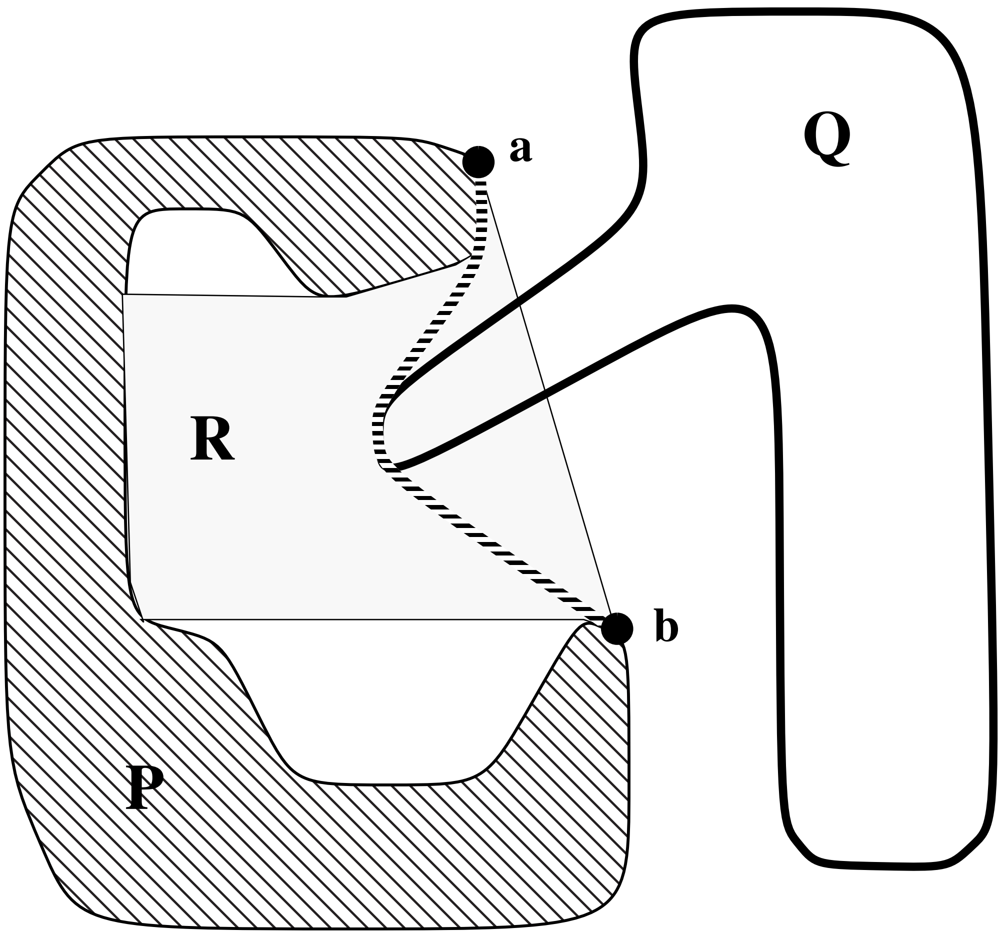

Figure 5: Illustrating the proof of theorem 1.

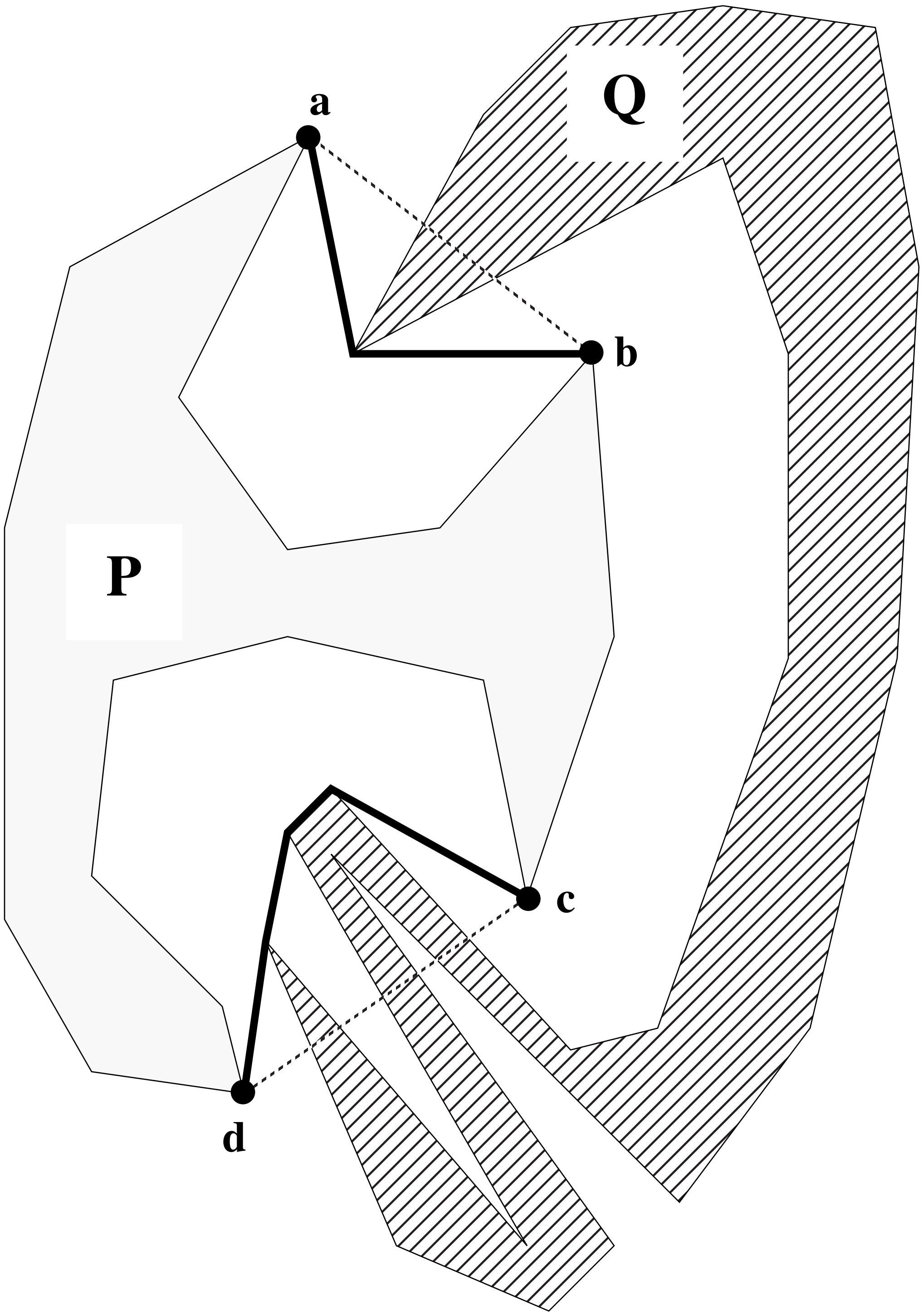

Figure 6: Illustrating one possible way to compute CH(P/Q).

relative hulls of P and Q respectively. If this were not the case then the relative hull of, say, P, CH(P | Q) would intersect either a pocket of VH(P,θ) or a pocket of VH(Q,θ). In either case it would imply that P and Q cannot be separated in direction θ, a contradiction. Now, the relative convex hulls of VH(P,θ) and VH(Q,θ) form convex polygons except for the chains lying inside pockets of the convex hulls of P and Q, respectively. Consider one such pocket determined by vertices *a* and *b* and refer to Fig. 5. The region R, where GP(*a, b* | P) must lie, is a monotonic polygon with respect to $\theta + 90^o$ and from Lemma 3.2 it follows that GP(*a, b* | P) is a monotonic chain with respect to $\theta + 90^o$. Since this is true for all such pockets we have that CH(P | Q) and CH(Q | P) are both monotonic with respect to $\theta + 90^o$. Q.E.D.

4. The Algorithm

Theorem 3.1 suggests the following algorithm for solving the separability problem stated in the abstract. Compute the relative convex hulls of P and Q and determine whether they are monotonic polygons. With an appropriate data structure to handle weakly-simple polygons, the algorithm of Preparata and Supowit [31] can determine whether the relative convex hulls are monotonic in O(n) time. Thus the crucial part of the problem is computing the relative convex hulls.

Consider the polygons P and Q in Fig. 6 and the convex hull of P, CH(P). It is clear that if Q did not intersect CH(P) then we would have CH(P | Q) = CH(P). This observation suggests an approach to computing CH(P | Q) by first determining CH(P) and subsequently patching up CH(P) at those pockets where Q intersects CH(P) to obtain CH(P | Q). All we need to do is compute the shortest path in each pocket of CH(P) from the endpoints of the pocket lids (such as (*a,b*) and (*c,d*) in Fig. 6) such that the shortest paths separate P from Q. However, it is not clear how to patch up all the pockets with a total complexity less than O(n log n). If for each pocket (such as the one determined by *cd* in Fig.6) we had a list of all the boundary points of Q intersected by line segment *cd* and, furthermore, if these intersection points were sorted along *cd*, then O(t(n)) time would suffice to compute CH(P | Q). A straightforward scan of these intersection points would isolate a simple polygon, call it R, inside the pocket of *cd* in which the geodesic path from *c* to *d* is guaranteed to lie. This scan has a complexity linear in k_1, the number of vertices of Q contained in the pocket of *cd*. The region R can then be triangulated in time $t(k_1 + k_2)$ where k_2 is the number of vertices of P contained in the pocket of *cd*. Finally, the geodesic path between *c* and *d* can be computed in $O(k_1 + k_2)$ time. Adding the time taken for all the pockets of P would lead to a complexity of O(t(n)). Unfortunately, it is not clear how to obtain all the intersection points in sorted order efficiently. Sorting Jordan sequences in linear time [28] does not appear to help. If for each pocket lid P we apply Jordan sorting to Q this results in an overall complexity of $O(n^2)$.

One way to reduce this complexity is by using the line-segment intersection algorithm of Mairson and Stolfi [29]. They have shown that given two sets of n line segments S_1 and S_2 such that the elements in each set are pairwise disjoint (their interiors do not intersect), all the intersecting pairs between S_1 and S_2 can be reported in O(n log n+I) time where I is the number of such pairs. Furthermore, for each line segment their algorithm reports all the intersection points in sorted order along the line segment. Now, in general, I can be $O(n^2)$ but in the problem considered here we have an additional structure that can be exploited. In our problem S_1 consists of the edges of Q and S_2 consists of the pocket lids of the pocket of CH(P). Furthermore, these lids form a *convex* polygon. Therefore each line segment of S_1 can intersect at most two line segments of S_2 and therefore I = O(n). Thus this approach solves the separability problem in O(n log n) time [37]. Although this is much better than $O(n^2)$ it is not an improvement over [19].

We are able to reduce the complexity to O(t(n)) by using an additional simple lemma. Refer to Fig.7.

Definition: A *bridge* of CH(P ∪ Q) is an edge of CH(P ∪ Q) joining a vertex of P to a vertex of Q. An endpoint of a bridge will be called a P-endpoint (resp. a Q-endpoint) if the endpoint is a vertex of P (resp. Q).

In general, a bridge B_i will connect some vertex p_u of P to some vertex q_v of Q. If p_u and q_v are the endpoints of B_i we highlight this fact by using the notation p_{ui} and q_{vi}. If the discussion is independent of the actual values of *u* and *v* we use the notation $p_{\bullet i}$ and $q_{\bullet i}$ to specify the endpoints of bridge B_i.

Lemma 4.1: [30]. Given two nonintersecting simple polygons P, Q, the convex hull CH(P ∪ Q) has either zero or two bridges.

Proof: Zero bridges result when one polygon lies in the interior of the convex hull of the other. Consider two consecutive bridges B_i, B_{i+1}. If B_i has a pair of endpoints ($p_{\bullet i}$, $q_{\bullet i}$), then B_{i+1} must have a pair ($q_{\bullet i+1}$, $p_{\bullet i+1}$). Therefore an *odd* number of bridges is impossible since we would have a chain of one polygon containing vertices of the other polygon. Therefore we can only have an *even* number of bridges. Now *two* can occur when P and Q are linearly separable. Assume we have an even number greater than two and consider two of these B_1 and B_2. The P-endpoints of B_1 and B_2 are connected by two chains of P and the Q-endpoints by two chains of Q. Therefore any other bridge B_i, $i>2$, implies P and Q intersect, which is a contradiction.

We are now ready to complete the description of the algorithm. In case 1, then, we have the CH(P ∪ Q) and two bridges $B_i = (q_{\bullet i}, p_{\bullet i})$ and $B_j = (p_{\bullet j}, q_{\bullet j})$ illustrated in Fig. 7. Note that the two bridges and the chains $C_{\bullet j \bullet i}(P)$ and $C_{\bullet i \bullet j}(Q)$ define a simple polygon Z(P, Q) which "separates" P and Q. The CH(P | Q) is the concatenation of the partial convex hull of P from $p_{.i}$ to $p_{.j}$ and the geodesic path in Z(P, Q) from $p_{\bullet j}$ to $p_{\bullet i}$. Similarly, the CH(Q | P) is the union of the partial hull of Q from $q_{\bullet j}$ to $q_{\bullet i}$ and the geodesic path in Z(P, Q) from $q_{\bullet i}$ to $q_{\bullet j}$. Since Z(P, Q) is simple, we can triangulate it in O(n log log n) time with the algorithm of Tarjan and Van Wyk [32]. Note that all computations other than triangulation are either linear or sublinear. The convex hulls of P and Q can be computed in O(n) time with the algorithm of McCallum and Avis [34] and detecting whether or not they intersect can be done in O(log n) time with the algorithm of Chazelle and Dobkin [35]. Finally, computing the convex hull of P ∪ Q ban be done in O(n) time using the "rotating caliper" algorithm of Toussaint [36].

In Case 2, when Q lies entirely within one pocket of P, the situation is slightly more involved. Let *ab* denote the pocket lid of some pocket K_j of P and refer to Fig. 8. The pocket K_j is itself a simple polygon determined by line segment *ab* concatenated with chain $C_{ba}(P)$. In this case the region of intersect in which we need to compute geodesic paths is the set-difference of K_j and the

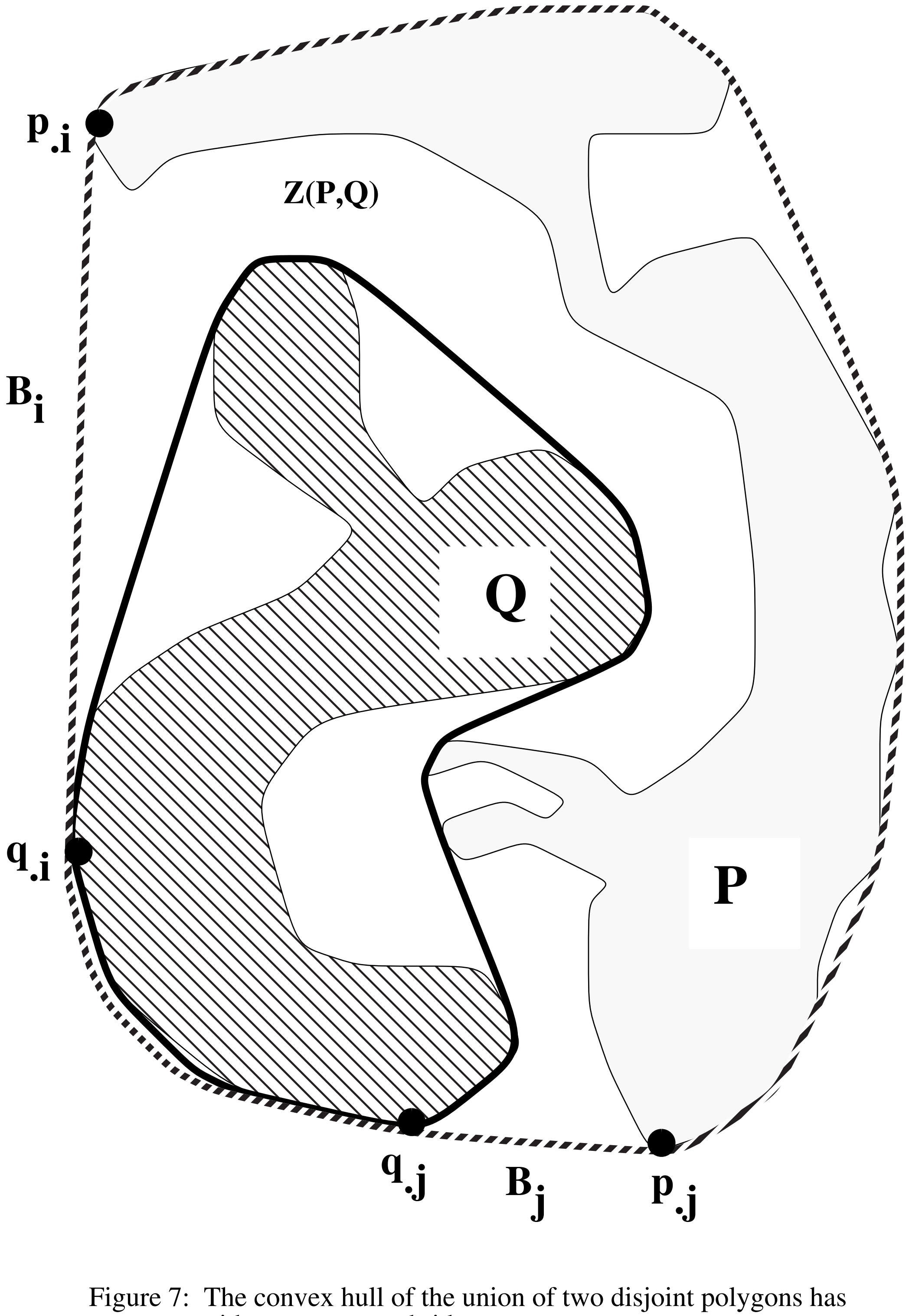

Figure 7: The convex hull of the union of two disjoint polygons has either *zero* or *two* bridges.

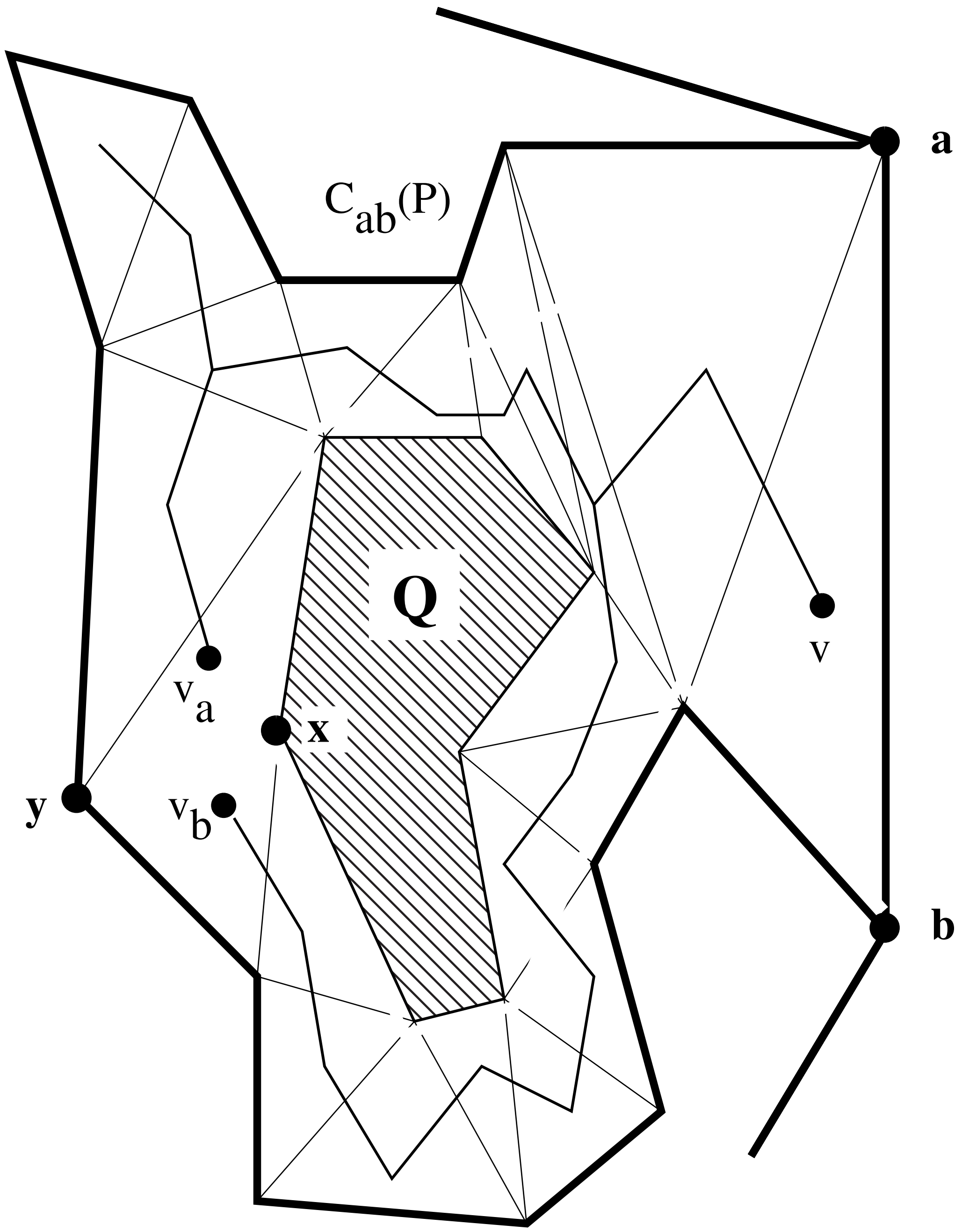

Figure 8: Illustrating the algorithm when Q lies entirely in the interior of one pocket of CH(P).

interior of Q. This region referred to as K'_j is not a simple polygon but contains a "hole." The first step is to convert K'_j into a *weakly simple* polygon P' by adding two "copies" of a "connecting bridge" *xy* between a vertex of Q and a vertex of $C_{ab}(P)$.

Definition: Given a pocket K_j of P properly containing polygon Q, a connecting *bridge* between Q and K_j is a line segment (x, y) such that (a) (x, y) does not intersect any edge of K_j or Q except at its endpoints, (b) $\mathrm{int}(x, y)$ lies in $\mathrm{int}(K_j)$ and in $\mathrm{ext}(Q)$, and (c) one endpoint of (x, y) is a vertex of Q and the other endpoint is a vertex of $C_{ab}(P)$ other than *a* and *b*.

Lemma 4.2: Given two polygons R_{out} and R_{in}, where R_{in} is properly contained in R_{out}, with n and m vertices, respectively, a connecting bridge between R_{in} and R_{out} can be found in $O(m + n)$ time.

Proof: The proof is straightforward and the details are omitted.

Once a bridge (x, y) between Q and $C_{ab}(P)$ has been found, P' can be triangulated and its dual tree T obtained. Three vertices of T play a singular role here. Let *v* be the node of T associated with the triangle having (a, b) as one of its edges. Let v_a and v_b be the nodes of T associated with the two triangles that share the connecting *bridge* (x, y). If $C_{ab}(P)$ is traversed in order starting at *a*, then an ordering of the triangles is induced. Of the two triangles sharing the connecting *bridge*, the first to be encountered in this ordering corresponds to v_a, the second to v_b. Although we cannot be sure that $CH(P \mid Q)$ must go through a specified point *z*, we do have the following lemma.

Lemma 4.3: $CH(P \mid Q)$ must intersect the connecting bridge (x, y).

Proof: By construction, K_j is a *simple* polygon. By definition, $CH(P \mid Q)$ in K_j, together with (a, b), is a *weakly simple* polygon, and in fact can be viewed as the relative convex hull $CH[Q \cup (a, b) \mid C_{ab}(P)]$ and therefore must contain Q. It follows that any *connecting bridge* must intersect $CH(P \mid Q)$. Q.E.D.

Lemma 4.3 now allows us to compute the geodesic path between *a* and *b* constrained to pass through the *connecting bridge xy* in linear time using the algorithms of Chazelle [25] or Lee and Preparata [23]. The algorithms in [25] and [23] compute the geodesic path between two points in a *sleeve* in linear time. A *sleeve* is a polygon whose dual tree is a chain. In our problem (see Fig. 8), the shortest path from v_a to *v* in T yields *sleeve* S_a. Similarly, the shortest path from v_b to *v* yields sleeve S_b. Unfortunately, the union of S_a and S_b where $GP(a, b \mid K'_j)$ must lie is not a simple polygon. However, we can get around this obstacle by embedding $S_a \cup S_b$ onto a Riemann surface [33] of two levels. We embed S_a onto level one and S_b onto level two with a *ramp* at the *connecting bridge xy* leading from level one to level two thus obtaining a *Riemann sleeve*. To the algorithms in [23] and [25] the *Riemann sleeve* so constructed looks just as if it were a regular sleeve. It follows that once P' is triangulated $CH(P \mid Q)$ can be computed in linear time.

All that remains is to compute $CH(Q \mid P)$. In this case it is easily verified that if we choose a *connecting bridge* (x, y) such that *x* is the vertex of Q furthest from the line through (a, b) then $CH(Q \mid P)$ must traverse *x*. Since the shortest path between v_a and v_b in T is a chain it follows that $GP(a', b' \mid P')$ where *a'* and *b'* are any two points lying in the triangles associated with v_a and v_b, respectively, must lie in a regular sleeve. Recall that in triangulating P' we first obtained a weakly simple polygon by inserting two copies of (x, y). Let x_a and x_b denote the two copies of x on the side of v_a and v_b, respectively. It follows that CH(Q | P) is GP(x_a, x_b | P') and can be computed as before.

In summary, we have shown:

Theorem 4.1: Given two disjoint simple polygons P and Q of n edges each, whether they are separable under translation can be determined in O(t(n)) time where t(n) is the time needed to triangulate a polygon with O(n) edges.

If P and Q are separable under translation it means CH(P | Q) and CH(Q | P) are monotonic with respect to some unoriented direction θ. Thus a motion for separation is immediate. Either P or Q can be translated in either of the two oriented directions determined by $\theta + 90^{\circ}$ and which of these orientations is valid can be determined in O(n) time. Actually, once the relative convex hulls are available in *all* directions in which P and Q are separable under translation can be computed in linear time. It suffices to know that the *wedge* of *all* possible directions is determined by the vertices of P (and Q), that (1) are contained in CH(Q) (CH(P)), and (2) that are also vertices of CH(P | Q) (CH(Q | P)). The details are omitted.

As a final note we remark that the new approach for solving the *translation-separability* problem presented here can lead to *optimal* algorithms if P and Q have additional structure that allows triangulation of the required regions to be done in linear time. As an example we obtain the following theorems by applying the previous results [37].

Theorem 4.2: Given two disjoint monotone polygons P and Q of n edges each, all motions that can take Q sufficiently far from P by a single translation can be determined in O(n) time.

Theorem 4.3: Given two disjoint star-shaped polygons P and Q of n edges each, all motions that can take Q sufficiently far from P by a single translation can be determined in O(n) time.

5. Conclusion

In closing we mention some open problems. The convex hull of a simple polygon can be found in linear time [34]. The relative convex hull seems very closely related to the standard convex hull and two polygons do not seem that much worse than one. Does there exist a linear algorithm for computing CH(P | Q) when P and Q are simple polygons? Two polygons may be interlocked under a single translation but not if rotations are allowed. How fast can we determine if two polygons can be separated with rotations? In three dimensions we may define CH(P | Q) for two polyhedra P and Q as the minimum-area surface enclosing P and excluding Q. How fast can we compute CH(P | Q)? Finally, there exists a family of problems that concern the "penetration" of the convex deficiency of P (the union of the pockets) by Q. For example, in the design of drugs we encounter the problem of finding a test molecule (modeled as a polyhedron) that will "fit" well "into" the deficiency of a host molecule. Several possibilities exist for measuring the degree of "penetration." One such measure might be the fraction of CH(P) taken up by CH(P) ~ CH(P | Q), where ~ denotes set difference. An open problem in both two and three dimensions is to determine for P and Q the *maximum penetration* under translations and rotations of Q without allowing collisions.

5. Acknowledgments

The author is grateful to David Avis and Hossam ElGindy for discussions on this topic and to Binay Bhattacharya for the suggestion that the region Z(P, Q) might help in reducing the complexity of this problem. As always, his intuition was unfailing.

6. References

[1] L. J. Guibas and F. F. Yao, On translating a set of rectangles, *Proc. Twelfth Annual ACM Symposium on Theory of Computing*, 154-160 (1980).

[2] T. Ottman and P. Widmayer, On translating a set of line segments, *Computer Vision, Graphics and Image Processing*, Vol. 24, 382-389 (1983).

[3] G. T. Toussaint, The complexity of movement, *Proc. IEEE International Symposium on Information Theory*, St. Jovite (September 1983).

[4] J. -R. Sack and G. T. Toussaint, Movability of objects, *Proc. IEEE International Symposium on Information Theory*, St. Jovite (September 1983).

[5] G. T. Toussaint and J. -R. Sack, Some new results on moving polygons in the plane, *Proc. Robotic Intelligence and Productivity Conference*, Detroit, Michigan, 158-163 (November 18-19, 1983).

[6] B. Chazelle *et al.*, The Complexity and decidability of SEPARATION, Technical Report CS-83-34, University of Waterloo (November 1983).

[7] G. T. Toussaint, On translating a set of polyhedra, *Proc. Conference on Polyhedra*, Smith College, Northhampton, Massachusetts (April 6-8, 1984).

[8] R. Dawson, On removing a ball without distributing the others, *Mathematics Magazine*, Vol. 57, No. 1, 27-30 (January 1984).

[9] K. Post, Six interlocking cylinders with respect to all directions, Internal manuscript, University of Eidenhoven (December 1983).

[10] W. Moser, Research problems in discrete geometry, Department of Mathematics, McGill University (1981).

[11] G. T. Toussaint, Some collision avoidance problems between spheres, *Proc. International Conference on Systems, Man and Cybernetics*, Tucson, Arizona (November 1985).

[12] Van-Duc Nguyen, The find-path problem in the plane, A. I. Memo, No. 160, Artificial Intelligence Laboratory, M.I.T. (February 1984).

[13] Bruce R. Donald, Hypothesizing channels through free-space in solving the findpath problem, A. I. Memo, No. 736, Artificial Intelligence Laboratory, M.I.T. (June 1983).

[14] David Marr and Lucia Vaina, Representation and recognition of the movement of shapes, A. I. Memo, No. 597, Artificial Intelligence Laboratory, M.I.T. (October 1980).

[15] G. T. Toussaint, Movable separability of sets, in *Computational Geometry*, G. T. Toussaint, Ed., North-Holland, Amsterdam, 335-375 (1985).

[16] S. Whitesides, Computational geometry and spatial planning, in *Computational Geo-*

metry, G. T. Toussaint, Ed., North-Holland, Amsterdam, 377-427 (1985).

[17] G. T. Toussaint and H. ElGindy, Separation of two monotone polygons in linear time, *Robotica*, Vol. 2, 215-220 (1984).

[18] J. -R. Sack and G. T. Toussaint, Translating polygons in the plane, *Proc. STACS'85*, Saarbrucken (January 1985).

[19] J. -R. Sack and G. T. Toussaint, Separability of pairs of polygons through single translations, *Robotica*, Vol. 5, 55-63 (1987).

[20] D. Kirkpatrick, Optimal search in planar subdivisions, *SIAM Journal on Computing*, Vol. 12, 28-35 (february 1983).

[21] R. Pollack, M. Sharir, and S. Sifrony, Separating two simple polygons by a sequence of translations, *Discrete and Computational Geometry*, Vol. 3, 123-136 (1988).

[22] C. Lantuejoul and F. Maisonneuve, Geodesic methods in quantitative image analysis, *Pattern Recognition*, Vol. 17, 1177-1187 (1984).

[23] D. T. Lee and F. P. Preparata, Euclidean shortest paths in the presence of rectilinear barriers, *Networks*, Vol. 14, 393-410 (1984).

[24] G. T. Toussaint, Shortest path solves edge-to-edge visibility in a polygon, *Pattern Recognition Letters*, Vol. 4, 165-170 (July 1986).

[25] B. Chazelle, A theorem on polygon cutting with applications, *Proc. 23rd Annual IEEE Symposium on Foundations of Computer Science*, 339-349 (1982).

[26] H. ElGindy, Hierarchical decomposition of polygons with applications, Ph.D. thesis, School of Computer Science, McGill University (May 1985).

[27] L. M. Kelly, Ed., *The Geometry of Metric and Linear Spaces*, Springer-Verlag, New York, (1985).

[28] K. Hoffman, K. Melhorn, P. Rosenstiehl, and R. E. Tarjan, Sorting Jordan sequences in linear time, *Proc. Symposium on Computational Geometry,* Baltimore, Maryland, 196-203 (June 1985).

[29] L. J. Guibas and J. Stolfi, Notes on computational geometry, Stanford University (1982).

[30] B. K. Bhattacharya and G. T. Toussaint, An O(n log log n) time algorithm for determining translation separability or two simple polygons, Technical Report No. SOCS-86.1, McGill University (January 1986).

[31] F. P. Preparata and K. J. Supowit, Testing a simple polygon for monotonicity, *Information Processing Letters*, Vol. 12, 161-164 (August 1981).

[32] R. E. Tarjan and C. J. Van Wyk, An O(n log log n) time algorithm for triangulating simple polygons, *SIAM Journal on Computing*, in press.

[33] J. Kahn, M. Klawe, and D. Kleitman, Traditional galleries require fewer watchmen, *SIAM Journal on Algebraic and Discrete Methods*, Vol. 4, 194-206 (June 1983).

[34] D. McCallum and D. Avis, A linear time algorithm for finding the convex hull of a simple polygon, *Information Processing Letters*, Vol. 8, 201-205 (1979).

[35] B. M. Chazelle and D. P. Dobkin, Detection is easier than computation, *Proc. 12th SI-*

GACT Symposium, Los Angeles, California (1980).

[36] G. T. Toussaint, Solving geometric problems with the rotating calipers, *Proc. MELECON'83*, Athens (May 1983).

[37] G. T. Toussaint, Shortest path solves translation separability of polygons, Technical Report SOCS-85.27, McGill University (October 1985).