[2] Schwartz, J. T.: Finding the minimum distance between two convex polygons. Information Processing Letters 1981, 168 - 170.

[3] Toussaint G. T., Bhattacharya, B. K.: Optimal algorithms for computing the minimum distance between two finite planar sets. Fifth International Congress of Cybernetics and Systems, Mexico City, August 1981.

[4] McKenna, M., Toussaint, G. T.: Finding the minimum vertex distance between two disjoint convex polygons in linear time. Tech. Report No.SOCS - 83.6, School of Computer Science, McGill University, April 1983.

[5] Chin, F., Wang, C. A.: Minimum vertex distance problem between two convex polygons. Technical report, University of Alberta, 1983.

[6] Supowit, K. J.: The relative neighborhood graph, with an application to minimum spanning trees. J.A.C.M. (in press).

[7] Lee, D. T., Preparata, F. P.: The all-nearest-neighbor problem for convex polygons. Information Processing Letters 7, 189 - 192 (1978).

[8] Yang, C. C., Lee, D. T.: A note on the all-nearest-neighbor problem for convex polygons. Information Processing Letters 8, 193 - 194 (1979).

[9] Chazelle, B.: Computational geometry and convexity. Ph. D. thesis, Department of Computer Science, Carnegie-Mellon University, July 1980.

[10] O'Rourke, J., *et al*.: A new linear algorithm for intersecting convex polygons. Computer Graphics and Image Processing 19, 384 - 391 (1982).

[11] Chazelle, B., Dobkin, D.: Detection is easier that computation. Proc. Twelfth Annual ACM Symposium on Theory of Computing, April 1980, pp. 146 - 153.

Since P_{nw} and Q_R are two linearly separable convex polygons $d_{min}(P_{nw}, Q_R)$ can be solved with the techniques of [4] and [5]. Thus we turn our attention to $d_{min}(P_{nw}, Q_L)$. We can decompose this problem into two subproblems by splitting Q_L into two convex polygons $Q_{L\text{-}out}$ and $Q_{L\text{-}in}$, whose vertices lie outside P_{nw} and inside P_{nw}, respectively. We can determine all the sub-chains of Q_L that lie inside and outside P_{nw}, and thus $Q_{L\text{-}out}$ and $Q_{L\text{-}in}$, by applying the simple linear algorithm of O'Rourke *et al*. [10] to intersect the two convex polygons P_{nw} and Q_L. We are left to solve for

$$d_{min}(P_{nw}, Q_L) = min\{d_{min}(P_{nw}, Q_{L\text{-out}}), d_{min}(P_{nw}, Q_{L\text{-in}})\}$$

Now $d_{min}(P_{nw}, Q_{L\text{-}out})$ is taken care of by theorem 2.1. Finally, since $Q_{L\text{-}in}$ lies completely inside P_{nw}, $d_{min}(P_{nw}, Q_{L\text{-}in})$ is nothing but case 1 revisited.

Therefore case 2 can also be solved in $O(m+n)$ time. It is possible to determine in $O(\log (m+n))$ time whether the interiors of P and Q intersect or not [11]. If the interiors intersect it is more difficult to determine whether one polygon is entirely inside another and, in fact, Chazelle [9] has proved an $\Omega(m+n)$ lower bound for this problem. However, using the linear intersection algorithm of O'Rourke *et al*. [10] we can solve this problem in $O(m+n)$ time by merely checking to see if all the vertices of $P \cap Q$ belong to only one of there polygons. We therefore have the following result.

Theorem 4.1: The minimum vertex-distance between two convex polygons P and Q of m and n vertices, respectively, can be computed in $O(m+n)$ time.

5. Open Problems

Several interesting problems remain. One pertains to three dimensions. Given two convex polyhedra in three dimensions is it possible to compute the minimum vertex distance in $o(mn)$ time. Another open question concerns the planar all-nearest-distance-between-sets problem. Here, given two convex polygons P and Q we want to find, in $O(m+n)$ time, for each vertex in P (or Q) the nearest vertex in Q (or P). In section two we saw a solution to this problem in the special case when one polygon has the semi-circle property and the other is "correctly" situated with respect to the first.

Finally, no linear algorithm exists for computing the Voronoi diagram of a convex polygon. In section 2 we saw how to compute, in linear time, the Voronoi diagram of a *semi-circle* polygon P_s within the region $RH(p_i, p_{i+1})$ outside P_s, where $\overline{p_i p_{i+1}}$ is the diameter of P_s. However, no linear algorithms exist for computing the Voronoi diagram in the interior of P_s or in $LH(p_i, p_{i+1})$. Such algorithms would allow us to solve the problem for arbitrary convex polygons since we can decompose a convex polygon into four *semi-circle* polygons in linear time and we can merge the four Voronoi diagrams in linear time.

6. References

[1] Edelsbrunner, H.: On computing the extreme distance between two convex polygons. Technical report F96, Technical University of Graz, 1982.

4. Case 2: P and Q Are Arbitrary Crossing Polygons

Let P and Q be two convex polygons arbitrarily placed. In this case the boundaries of P and Q may have as many as $m + n$ proper intersection points. We will exhibit a decomposition of this problem into at most 12 subproblems such that each of these can be solved by either the algorithms in [4] and [5], theorem 2.1 of this paper, or the procedure for case 1.

Problem decomposition

Step 1: This step is identical to step 1 for case 1: Thus we must solve four problems now of the form $d_{min}(P_{nw}, Q)$. (See Fig. 4.) We decompose this problem further into 3 subproblems.

Step 2: Draw a line L through p_{xmin} and p_{ymax} and determine the intersections that L makes with Q as before. L partitions Q into two polygons, as before, Q_R and Q_L and

$$d_{min}(P_{nw}, Q) = min\{d_{min}(P_{nw}, Q_R), d_{min}(P_{nw}, Q_L)\}$$

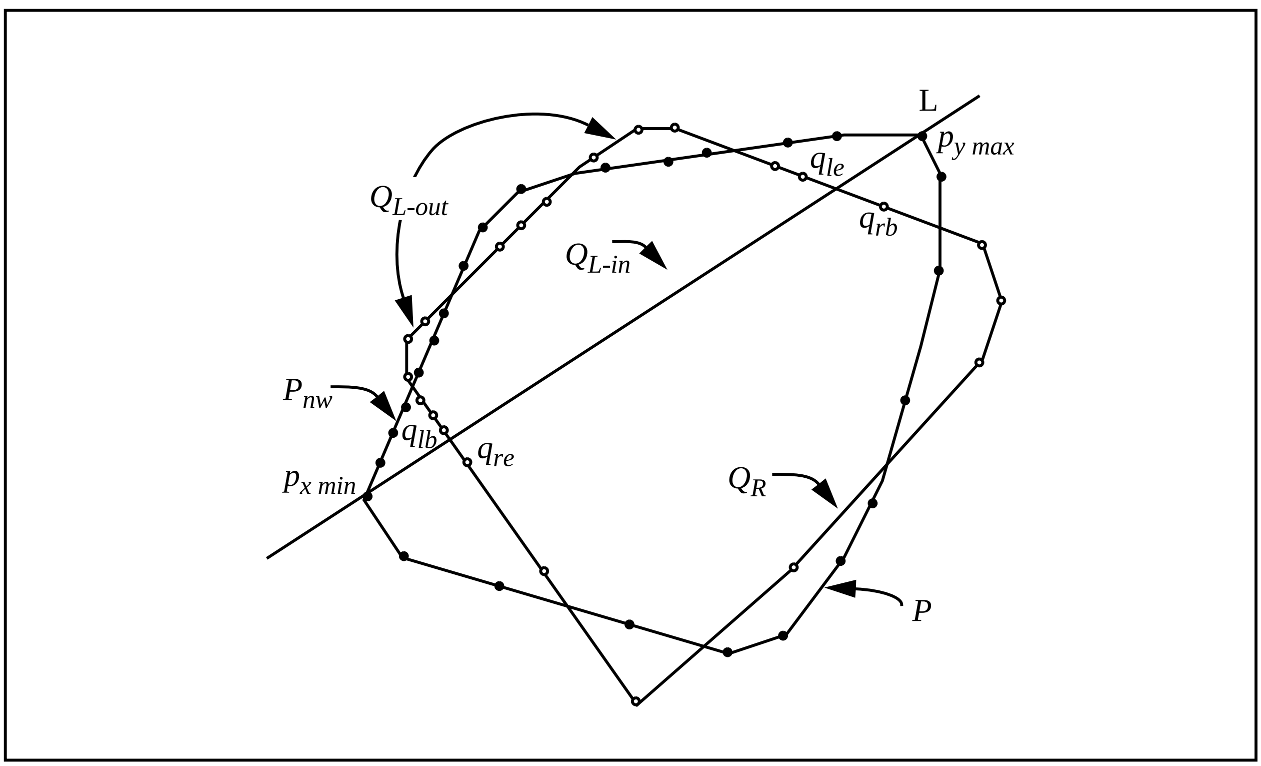

Fig. 4.

$$d_{min}(P, Q) = min\{d_{min}(P_{ne}, Q), d_{min}(P_{se}, Q), d_{min}(P_{sw}, Q), d_{min}(P_{nw}, Q)\}$$

and therefore we need only solve four problems of the form $d_{min}(P, Q)$, i.e., a *semi-circle* polygon lying completely inside a convex polygon. We will further decompose each such problem into two subproblems as follows:

Step 2: Draw a line L through p_{ymax} and p_{xmin} and determine the intersection points of L with the boundary of Q. This can be done in O(log n) time with an algorithm of Chazelle [9]. Without loss of generality assume L is vertical for convenience and refer to Fig. 3. The line L partitions the plane into two half planes $RH(p_{xmin}, p_{ymax})$ and $LH(p_{xmin}, p_{ymax})$. It also partitions Q into two convex polygons $Q_l = (q_{lb},..., q_{le})$ and $Q_r = (q_{rb},..., q_{re})$, where $\overline{q_{le}q_{rb}}$ and $\overline{q_{re}q_{lb}}$ are the edges of Q intersected by L. Note that if L intersects some vertex q_i of Q then we may have $q_{le} = q_i = q_{rb}$. Furthermore $Q_L \in LH(p_{xmin}, p_{ymax})$ and $Q_R \in RH(p_{xmin}, p_{ymax})$. We now have

$$d_{min}(P_{nw}, Q) = min\{d_{min}(P_{nw}, Q_L), d_{min}(P_{nw}, Q_R)\}$$

To solve for $d_{min}(P_{nw}, Q_l)$ we can invoke theorem 2.1. Finally, since P_{nw} and Q_R are linearly separable, $d_{min}(P_{nw}, Q_R)$ can be solved with the techniques of [4] and [5]. Therefore case 1 can be solved in O(m+n) time.

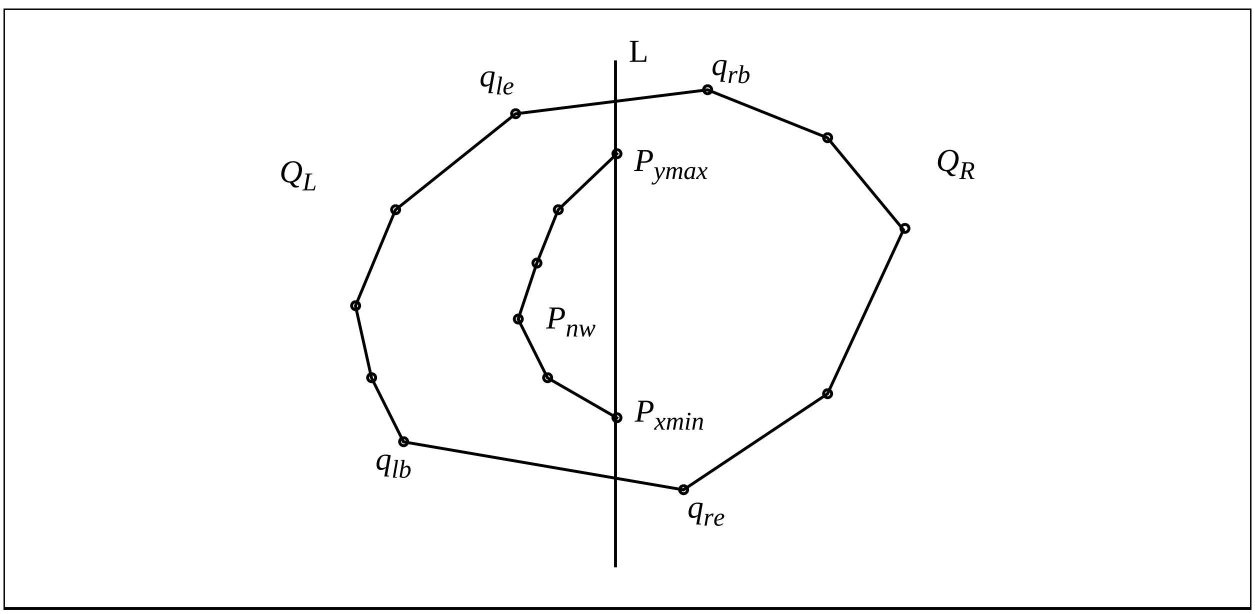

Fig. 3.

3. Case 1: P Lies Entirely Inside Q

Without loss of generality let us assume that P lies inside Q, i.e., $P \cup Q = Q$. We will decompose this problem into at most eight subproblems, four of which are linearly separable and can be solved with the techniques of [4] and [5], and four which are taken care of by theorem 2.1 in this paper. First we decompose P into four *semi-circle* polygons. Both Lee and Preparata [7] and Yang and Lee [8] give O(m) algorithms for obtaining such a decomposition. We select the latter [8] because it is simpler and does not require the computation of the diameter as in [7].

Problem decomposition

Step 1: Find p_{xmax}, p_{xmin}, p_{ymax}, and p_{ymin}, the vertices of P with extreme x and y coordinates. (See Fig. 2.) We then obtain four convex polygons with the *semi-circle* property:

$$P_{ne} = (p_{ymax}, \ldots, p_{xmax})$$
$$P_{se} = (p_{xmax}, \ldots, p_{ymin})$$
$$P_{sw} = (p_{ymin}, \ldots, p_{xmin})$$
$$P_{nw} = (p_{xmin}, \ldots, p_{ymax})$$

Note that if two vertices have the same coordinate, for example p_{ymax}, then the left vertex is associated with P_{nw} and the right vertex with P_{ne} and so on.

Now, denoting the minimum vertex distance between P and Q by $d_{min}(P, Q)$, we have that

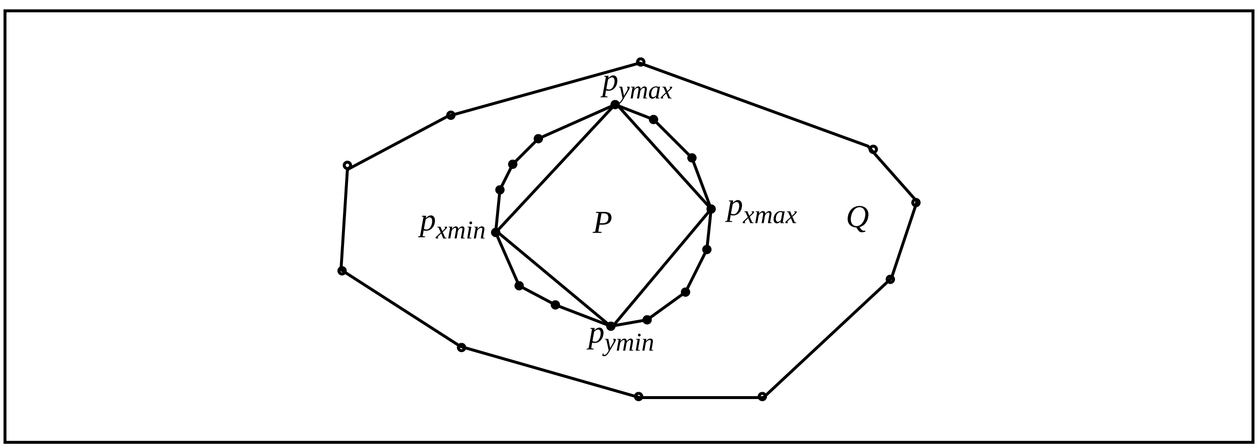

Fig. 2.

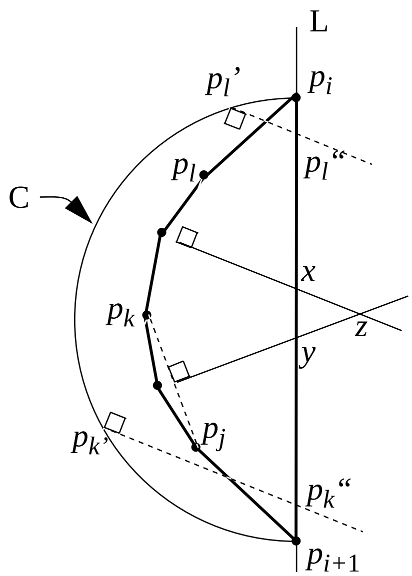

Fig.1.

p_k' must intersect $\overline{p_i p_{i+1}}$ at p_k''. Thus it follows that $RB(p_k, p_l)$ must intersect $\overline{p_i p_{i+1}}$ at some point, say x. Similarly, the ⊥ bisector of $\overline{p_j p_k}$ must intersect $\overline{p_i p_{i+1}}$ at some point say y. Furthermore, since *angle* $p_j p_k p_l < 180°$ the intersection of $RB(p_j, p_k)$ and $RB(p_k, p_l)$, say z, must lie in $RB(p_j, p_k) \cap RB(p_k, p_l)$. If x lies above y then $z \in RB(p_{i+1}, p_i)$ and we are done. If x lies below y, then $z \in LH(p_{i+1}, p_i)$ and we must show that $z \in P_s$. Therefore assume x lies below y. Construct triangles $\Delta x p_k p_l \equiv \Delta_x$ and $\Delta y p_j p_l \equiv \Delta_y$. From convexity it follows that $\Delta_x \in P_s$ and $\Delta_y \in P_s$. Therefore, the portion of $RB(p_k, p_l)$ to the left of L lies in P_s and also the portion of $RB(p_j, p_k)$ to the left of L lies in P_s. Therefore $z \in P_s$. Since the triplet p_j, p_k, p_l was arbitrary it follows that all $O(n^3)$ *local* Voronoi vertices of P_s lie in P_s or $RH(p_{i+1}, p_i)$. Since a Voronoi vertex of $VD(P_s)$, or *global* Voronoi vertex, belongs to a subset of the *local* vertices it follows that all $O(n)$ Voronoi vertices of $VD(P_s)$ lie in P_s or $RH(p_{i+1}, p_i)$. Q.E.D.

This theorem implies that the Voronoi diagram of P_s in the region to the left of L and exterior to P_s is completely determined by the partition imposed by the ⊥ bisectors of the edges $\overline{p_{i+1} p_{i+2}}, \overline{p_{i+2} p_{i+3}}, \ldots, \overline{p_{i-1} p_i}$. Therefore, in this region the Voronoi diagram can be constructed in $O(n)$ time. Furthermore, the "layered" structure of the Voronoi diagram implies that n query points forming a convex chain $CQ = (q_1, q_2, \ldots, q_n)$, such that its vertices lie in such a region, can be searched for point location in a total running time of $O(n)$. Thus for this special situation the nearest point P_s to each point in CQ can be solved in $O(n)$ time. It follows that the minimum-vertex-distance in between CQ and P_s can be computed in $O(n)$ time.

is based on existing results on the relative neighborhood graph [6]. With trivial modifications the algorithms in [4] and [5] also work if only the edges of P and Q intersect, i.e., as long as the interiors of the polygons do not intersect.

In this paper we show that when the interiors of P and Q intersect the minimum vertex distance can also be computed in O($m+n$) time. The problem is split into two cases: the case when one polygon is completely contained in the other and the case where this is not true. The key result for obtaining a solution to both cases consists of decomposing a convex polygon into parts associated with regions on the plane where the Voronoi diagram can be computed in linear time. This result is presented in section 2. Section 3 describes the algorithm for the case when one polygon is contained in the other and the case where this is not true is treated in section 4. Finally section 5 discusses some open problems.

2. Preliminary Results

Lee and Preparata [7] obtained a linear-time algorithm for the all-nearest-neighbor problem for a convex polygon P by decomposing P into four *semi-circle* polygons. Consider the following conditions:

(i) The two farthest points of P are the extremes of an edge, i.e., diameter(P) = $d(p_i, q_{i+1})$ for some i.

(ii) All the vertices of P lie inside a circle with diameter equal to the diameter of P. A convex polygon that satisfies both (i) and (ii) is a *semi-circle* polygon.

Semi-circle polygons have some very special properties. The property used in [7] is the fact that for any vertex p_i its nearest neighbor p_j is adjacent to p_i, i.e., it is either p_{i+1} or p_{i-1}. In this section we prove another special property of *semi-circle* polygons. They admit a partition of the plane into special regions, needed for solving the minimum vertex-distance problem, where the Voronoi diagram can be constructed in linear time. Furthermore, this Voronoi diagram can be searched for point location of a linear number of query points in linear time when the query points are vertices of a convex polygonal chain.

Let $L(p_i, p_j)$ denote the directed straight line through p_i and p_j in that order. Let $RH(p_i, p_j)$ denote the closed half-plane lying to the right of $L(p_i, p_j)$, i.e., it includes $L(p_i, p_j)$. If it does not include the line it will be referred to as open. Also LH will refer to left half-plane. Let $VD(P)$ denote the Voronoi diagram of the vertices of P, $B(p_i, p_j)$ the perpendicular ($\perp$) bisector of the line segment joining p_i and p_j, and let $RB(p_i, p_j)$ denote that part of $B(p_i, p_j)$ lying to the right of $L(p_i, p_j)$.

Theorem 2.1: Given a convex polygon P_s of n sides with the semi-circle property with respect to edge $\overline{p_i p_{i+1}}$ then the Voronoi vertices of $VD(P_s)$ all lie in P_s or in open $RH(p_{i+1}, p_i)$.

Proof: Without loss of generality, we assume $\overline{p_i p_{i+1}}$ is vertical. Let p_j, p_k, p_l be any ordered triplet of vertices of P_s. The local Voronoi vertex of this triplet v_{jkl} is determined by the intersections of the $\perp$ bisector of $\overline{p_j p_k}$ and $\overline{p_k p_l}$. Extend $\overline{p_k p_l}$ to intersect the semi-circle C at p_l' and extend $\overline{p_l p_k}$ to intersect at C at p_k'. (Refer to Fig. 1.) Since *angle* $p_k p_l p_i \geq 90°$ it follows that the $\perp$ to $L(p_k, p_l)$ at p_l' intersects $\overline{p_i p_{i+1}}$ at p_l''. Since $p_{i+1} p_k p_l \geq 90°$, the $\perp$ to $L(p_k, p_l)$ at

An Optimal Algorithm for Computing the Minimum Vertex Distance Between Two Crossing Convex Polygons*

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

ABSTRACT

Let $P = \{p_1, p_2,..., p_m\}$ and $Q = \{q_1, q_2,..., q_n\}$ be two intersecting polygons whose vertices are specified by their cartesian coordinates in order. An optimal O(m+n) algorithm is presented for computing the minimum euclidean distance between a vertex p_i in P and a vertex q_j in Q.

Key words: Algorithms, complexity, computational geometry, convex polygons, minimum distance, Voronoi diagrams.

1. Introduction

Let P = $\{p_1, p_2,..., p_m\}$ and $Q = \{q_1, q_2,..., q_n\}$ be two convex polygons whose vertices are specified by their cartesian coordinates in clockwise order. We assume the polygons are in *standard* form, i.e., no three vertices are collinear. Let $d(x, y)$ denote the euclidean distance between points x and y. Considerable attention has been given recently to the problem of computing extremal distances between convex polygons due to their application in pattern recognition and collision avoidance problems [1], [2]. One such problem consists of finding the *minimum* distance between the polygons, i.e., zero if the polygons intersect and the minimum distance d(x, y) realized by a pair of points $x \in P$, $y \in Q$, if P and Q do not intersect. Edelsbrunner [1] describes an optimal O(log m + log n) algorithm for solving this problem. This improves an earlier algorithm for this problem due to Schwartz [2] which runs in O((log m)(log n)) time.

A more difficult problem is to find the *minimum vertex distance* between P and Q, i.e., the minimum distance $d(x, y)$ where x and y are restricted to being vertices of P and Q, respectively. The naive method of computing $d(p_i, q_j)$ for all i and j requires, of course, O(mn) time. By computing supergraphs of the minimal spanning tree of the union of the vertices of P and Q Toussaint and Bhattacharya [3] have shown that this problem can be solved in O((m+n) log (m+n)) time. The methods of [3] however do not exploit the fact that P and Q are convex.

Recently McKenna and Toussaint [4] and Chin and Wang [5] independently discovered optimal O(m+n) algorithms for solving this problem in the special case where P and Q are *linearly separable*, i.e., the polygons do not intersect. The algorithm in [4] differs from that in [5] in that it

* Research supported by NSERC grant no. A9293.