Dynamic circle separability between convex polygons

Luis Barba[1], **Jorge Urrutia**[2]

[1] Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México
l.barba@uxmcc2.iimas.unam.mx

[2] Instituto de Matemáticas, Universidad Nacional Autónoma de México
urrutia@matem.unam.mx

Abstract. Let P, Q be two polygons of n and m vertices, respectively. A circle containing P and whose interior does not intersect Q is called a *separating circle*. We propose an algorithm for finding the minimum separating circle between a fixed convex polygon P and query convex polygon Q. The polygons P and Q are given as ordered lists of vertices (sorted according to their order of appearance along the convex hulls of P and Q respectively). We perform a linear time preprocessing on the number of vertices of P; the query time complexity is $O(\log n \log m)$.

Introduction

Kim and Anderson [**1**] presented a quadratic algorithm for solving the circular separability problem between any two finite planar sets. Bhattacharya [**2**] improved the running time to $O(n \log n)$. Finally O'Rourke, Kosaraju and Megiddo [**3**] found an optimal linear time algorithm to solve this problem. In this paper we study a new version of the problem. Let P be a fixed convex polygon with n vertices. We propose an algorithm for solving the circular separability problem between P and any query convex polygon Q with m vertices, both given as an ordered list of their elements. Our algorithm uses a linear time preprocessing on P, and has $O(\log n \log m)$ query time complexity.

1 Circular separability

Suppose for ease of description that the vertices of P and Q are in general position, and that P has no four co-circular vertices. Let C_P be the minimum enclosing circle of P and let c_P be its center. It is known that c_P can be found in $O(n)$ time [**4**]. Note that c_P is a point on an edge of the farthest-point Voronoi diagram of the vertices of P. Clearly if the interiors of Q and P are not disjoint, our problem has no solution, hence we will suppose that $d(P, Q) \geq 0$. It is also clear that if Q and C_P have disjoint interiors, then C_P is trivially the minimum separating circle.

1.1 Preprocessing

We first calculate the farthest-point Voronoi diagram of the vertices of P in linear time [**5**]. It can be seen as a tree rooted in c_P and created by adding leaves on every unbounded edge; we will denote this tree as $\mathcal{V}(P)$. For each vertex p of P, let $R(p)$ be the farthest-point Voronoi region associated to p, and assume that p has a pointer to $R(p)$. Let x be a point on an edge of $\mathcal{V}(P)$, and let T_x denote the path contained in $\mathcal{V}(P)$ joining c_P to x.

We will use the data structure on $\mathcal{V}(P)$ proposed by Roy, Karmakar, Das and Nandy in [**6**], which can be constructed in linear time and uses linear space. Given a vertex v in the tree $\mathcal{V}(P)$, this data structure allows us to do a binary search on the vertices of $\mathcal{V}(P)$ lying on T_v.

1.2 The minimum separating circle

We will call every circle containing P and whose interior does not intersect Q a *separating circle*. Let c' be the center of the minimum separating circle. In this section we will find c' starting from the center of an arbitrary separating circle.

Given $x \in \mathbb{R}^2$, let $C(x)$ be the minimum enclosing circle of P with center on x, and let $\rho(x)$ be the radius of $C(x)$. The following is a well-known result for the farthest-point Voronoi diagram.

Proposition 1.1. *Let x be a point on $\mathcal{V}(P)$. Then ρ is a monotonically increasing function along the path T_x starting at c_P.*

We now address some properties of separating circles, some of which are given without proof.

Observation 1.2. *The minimum separating circle has its center on $\mathcal{V}(P)$.*

Observation 1.3. *Let $x, y \in \mathbb{R}^2$. For every $z \in [x, y]$, we have $C(z) \subseteq C(x) \cup C(y)$.*

The previous observation implies that the minimum separating circle is unique.

Proposition 1.4. *Let x, y be two points on $\mathcal{V}(P)$ such that $C(x), C(y)$ are separating circles and x, y belong to the boundary of the Voronoi region $R(p)$. If z is the lowest common ancestor of x and y in $\mathcal{V}(P)$, then $C(z)$ is a separating circle; moreover, we have $\rho(z) \leq \min\{\rho(x), \rho(y)\}$.*

Proof. Suppose that $y \notin T_x$ and $x \notin T_y$; otherwise the result follows trivially. Assume then that the paths connecting x and y to z have disjoint relative interiors. Let $\ell_{z,p}$ be the straight line through z and p; this line leaves x and y in different semiplanes. Let z' be the intersection between $\ell_{z,p}$ and $[x, y]$. By Observation 1.3 we know that $C(z') \subseteq C(x) \cup C(y)$. Since z', z, p are co-linear, we have $C(z) \subseteq C(z')$ and thus $\rho(z) < \rho(z')$; see Figure 1(a). Finally, by transitivity we have that $C(z) \subset C(x) \cup C(y)$, which implies that $C(z)$ is a separating circle. Using Proposition 1.1, we conclude that $\rho(z) \leq \min\{\rho(x), \rho(y)\}$. $\square$

Now we generalize the previous result.

Lemma 1.5. *Let x, y be two points on $\mathcal{V}(P)$ such that $C(x), C(y)$ are separating circles. If z is the lowest common ancestor of x and y in the rooted tree $\mathcal{V}(P)$, then $C(z)$ is a separating circle; moreover, $\rho(z) \leq \min\{\rho(x), \rho(y)\}$.*

Proof. Proceeding by contradiction, suppose that $C(z)$ is not a separating circle. Let w_x be a point on T_x such that $\rho(w_x) = \min\{\rho(w) : w \in T_x \text{ and } C(w) \text{ is a separating circle}\}$; thus $w_x \neq z$. Consider the intersections of the segment $[w_x, y]$ with $\mathcal{V}(P)$ and suppose that the intersection points are $w_x = x_0, x_1, \ldots, x_k = y$ in this order. Let z' be the lowest common ancestor of w_x and x_1 in $\mathcal{V}(P)$. It is clear that w_x and x_1 belong to the same Voronoi region. Thus, by Proposition 1.4, $C(z')$ is a separating circle. Note that z' belongs to T_x, which is a contradiction with the definition of w_x. Our result follows. $\square$

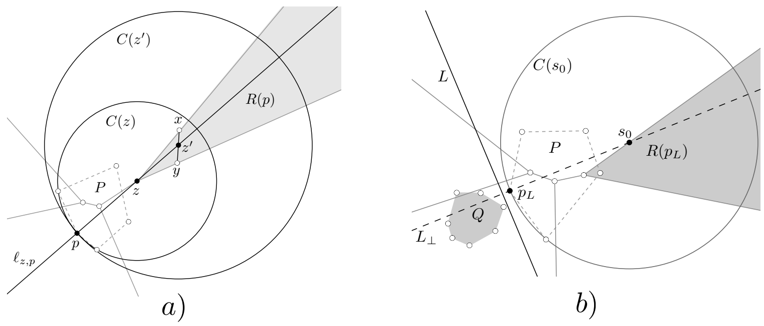

FIGURE 1. (a) Proof of Proposition 1.4. (b) The construction of s_0.

Theorem 1.6. *Let s be a point on an edge of $\mathcal{V}(P)$ such that $C(s)$ is a separating circle. Then c' belongs to T_s.*

Proof. Let w be a point on an edge of T_s such that

$$\rho(w) = \min\{\rho(z) \mid z \in T_s \text{ and } C(z) \text{ is a separating circle}\}.$$

Suppose that $w \neq c'$; thus $c' \notin T_s$. Therefore, by Lemma 1.5, if z is the lowest common ancestor of c' and w, then $C(z)$ is a separating circle with $\rho(z) \leq \rho(c')$. Also, since $c' \notin T_w \subseteq T_s$, the inequality is strict, which is a contradiction; our result follows. □

2 The algorithm

In this section, we present an algorithm to find c'. Our algorithm first finds a separating circle with center s_0 on an edge of $\mathcal{V}(P)$. Then we search for c' using a binary search on T_{s_0}.

We first construct a straight line L separating P and Q in logarithmic time [**7**]. Let us assume that p_L is the unique point in P closest to L. Otherwise, rotate L slightly, keeping P and Q separated by L. Let $L_\perp$ be the perpendicular to L that contains p_L and let s_0 be the intersection of $L_\perp$ with the boundary of $R(p_L)$. Note that $d(s_0, p_L)$ defines the radius of $C(s_0)$, therefore $C(s_0)$ is a separating circle; see Figure 1(b). Also, by construction s_0 is on an edge of $\mathcal{V}(P)$. It is clear that we can find s_0 in $O(\log n + \log m)$ time. Suppose that s_0 is on the edge xy of $\mathcal{V}(P)$, and let $T_x = (c_P = u_0, u_1, \dots, u_{r-1} = y, u_r = x)$. It follows from Theorem 1.6 that c' is on an edge of T_x.

Using the data structure proposed by Roy, Karmakar, Das and Nandy [**6**], we perform a binary search for c' on the vertices of T_x as follows. Initially, let $j = 0$, and $k = r$. Let u_i be the mid-vertex on the path of T_x between u_j and u_k. First compute $d(u_i, Q)$ in $O(\log m)$ time [**7**]. Now, in constant time, calculate $\rho(u_i)$. If $d(u_i, Q) = \rho(u_i)$, then $u_i = c'$ and the algorithm ends. If $d(u_i, Q) < \rho(u_i)$, then we search for c' between u_i and u_k; if $d(u_i, Q) > \rho(u_i)$, then we search for c' between u_j and u_i.

Two possibilities arise. If c' is a vertex on $\mathcal{V}(P)$, then we will find it in $O(\log n)$ steps. Otherwise, if c' is an interior point of an edge $S = [u, v]$ of $\mathcal{V}(P)$, our algorithm will return S such that $c' \in S$. Since each step of the binary search requires $O(\log m)$ time, the complexity of the previous search is $O(\log n \log m)$.

Suppose that S is contained in the bisector of two vertices p_0, p_1 of P, and let Q_S be the set of points on the boundary of Q visible from every point in S. It can be computed in $O(\log m)$ time. Let $q_{c'}$ be the point of intersection of $C(c')$ and Q. Clearly $q_{c'}$ belongs to Q_S; see Figure 2(a). Given three points $p, q, r \in \mathbb{R}^2$, let $C(pqr)$ be the circumcircle of the triangle $\triangle(pqr)$. For $x \in Q_S$, let $F(x)$ be the radius of the circle $C(p_0 x p_1)$. It is easy to see that $F(x)$ is unimodal on Q_S and attains its maximal at $q_{c'}$; see Figure 2(b).

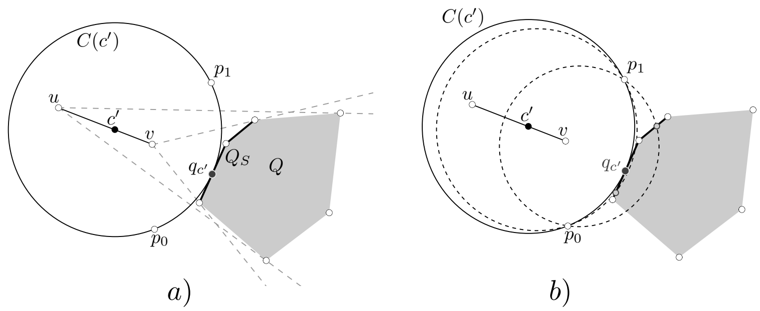

FIGURE 2. (a) The construction of Q_S. (b) $q_{c'}$ is maximal under F.

Let $Q_S^* = \{q_0, q_1, \ldots, q_r\}$ be the set of vertices of Q lying on Q_S. We can perform a binary search for $q_{c'}$ on the sorted list Q_S^* as follows. At each step we take the midpoint q^* of the current search list (initially Q_S^*), and compute the value of $F(q^*)$ in constant time. Take two points on each side of q^* at epsilon distance on the boundary of Q. If q^* is a local maximum of F, then the algorithm returns $q_{c'} = q^*$. Otherwise, determine if $q_{c'}$ lies to the left or to the right of q^*. Eliminate half of the list according to the position of $q_{c'}$ and repeat recursively. Our algorithm returns either the value of $q_{c'}$ if it is a vertex of Q, or a segment $H = (q_i, q_{i+1})$ of Q_S such that $q_{c'}$ belongs to H. In the first case, we are done, since c' can be determined in constant time given the position of $q_{c'}$. In the second case, the problem is reduced to that of finding a point $c' \in S$ such that $d(c', p_0) = d(c', H)$. This case can be solved with a quadratic equation in constant time.

Since each step of the binary search requires constant time, the algorithm finds the point $q_{c'}$ in $O(\log m)$ time, giving an overall complexity of $O(\log n \log m)$ for the algorithm.

References

[1] C. E. Kim and T. A. Anderson, Digital disks and a digital compactness measure, *STOC'84: Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, 1984, 117–124.

[2] B. K. Bhattacharya, Circular separability of planar point sets, *Computational Morphology* (1988), 25–39.

[3] J. O'Rourke, S. R. Kosaraju and N. Megiddo, Computing circular separability, *Discrete and Computational Geometry* **1** (1986), 105–113.

[4] N. Megiddo, Linear-time algorithms for linear programming in $\mathbb{R}^3$ and related problems, *SIAM Journal on Computing* **12** (1983), 759–776.

[5] A. Aggarwal, L. J. Guibas, J. Saxe and P. W. Shor, A linear-time algorithm for computing the Voronoi diagram of a convex polygon, *Discrete and Computational Geometry* **4** (1989), 591–604.

[6] S. Roy, A. Karmakar, S. Das and S. C. Nandy, Constrained minimum enclosing circle with center on a query line segment, *Computational Geometry* **42** (2009), 632–638.

[7] H. Edelsbrunner, Computing the extreme distances between two convex polygons, *Journal of Algorithms* **6** (1985), 213–224.