MAXIMIN LOCATION OF CONVEX OBJECTS IN A POLYGON AND RELATED DYNAMIC VORONOI DIAGRAMS

Keiko Imai
Chuo University

Hiroshi Imai
the University of Tokyo

Takeshi Tokuyama
IBM Tokyo Research Laboratory

Abstract   This paper considers the maximin placement of a convex polygon P inside a polygon Q, and introduces several new static and dynamic Voronoi diagrams to solve the problem. It is shown that P can be placed inside Q, using translation and rotation, so that the minimum Euclidean distance between any point on P and any point on Q is maximized in $O(m^4 n \lambda_{16}(mn) \log mn)$time, where m and n are the numbers of edges of P and Q, respectively, and $\lambda_{16}(N)$ is the maximum length of Davenport-Schinzel sequences on N alphabets of order 16. If only translation is allowed, the problem can be solved in $O(mn \log mn)$ time. The problem of placing multiple translates of P inside Q in a maximin manner is also considered.

1. Introduction

The polygon containment problem is to place a given polygon P inside another given polygon Q. This problem is closely related to the motion planning problem, and has been investigated extensively (e.g., Avnaim, Boissonnat [6], Chazelle [7], Chew, Kedem [8], Fortune [9], Leven, Sharir [14]).

In this paper, we consider several maximin polygon containment problems, which are stated very naturally as follows. The most fundamental maximin problem is to place a convex polygon P inside another polygon Q so that the minimum Euclidean distance between any point of P and any point of Q is maximized. Further, we consider the problem of placing multiple translates of P horizontally at regular intervals inside Q in a similar maximin way. Intuitively, the polygon P or its copies are placed inside Q so that they are as far from the boundary of Q as possible. A similar observation would hold for the problem of locating the largest similar copy of P in Q, which has been considered by Fortune [9], Leven and Sharir [14], Chew and Kedem [8], but they use the convex distance function which might not be good when P is thin. Although measuring the distance between P and Q by the Euclidean distance is very natural, there seems to have been done no research using the Euclidean distance.

We present efficient algorithms for these maximin location problems by introducing a new Voronoi diagram and its dynamic version for moving objects, and analyzing their combinatorial complexity. In analyzing the complexity, the theory of Davenport-Schinzel sequences is utilized (e.g., [1], [2], [5], [15]). We also investigate the combinatorial complexity of fundamental Voronoi diagrams for k rigidly moving sets of n points, which is really a canonical case and whose result is of interest by itself, and then apply the techniques used in this analysis to the new diagram.

The above problems arise in placing regional names on a map nicely, which is a major step toward realizing a good user interface, called the semantic overview function, in geographical databases (Aonuma, Imai, Kambayashi [4]). Here, the name may be represented by a rectangle and the region by a polygon, and we may place the rectangle inside the polygon in

the above-mentioned maximin manner. When the names, say, of schools have been placed at their sites in a region, the characters of the regional name may be represented as the congruent rectangle, and they may be placed horizontally at regular intervals in the region without making any intersections with the already placed characters. This corresponds to placing translates of a rectangle at regular intervals inside a polygonal region with holes. See Figure 1. The Voronoi diagrams introduced in this paper may also be used in the collision avoidance problem in robotics, especially to find a high-clearance path.

We now describe several maximin placement problems and present our results for each problem, comparing them with existing results. We are given a convex polygon P of m vertices and a polygon (or a polygonal region) Q of n vertices.

(P1) *Locate the polygon P, using translation only, inside the polygonal region Q so that the minimum Euclidean distance between any point on P and any point on Q is maximized.*

Problem (P1) is related to locating the largest similar copy of P inside Q by translation only, which can be done in $O(mn \log mn)$ time by using the Voronoi diagram of Q for the convex distance function concerning P [9]. In this paper, we define a new Voronoi diagram related to (P1), and show that this diagram can be constructed in $O(mn \log mn)$ time by combining the Voronoi-diagram algorithm [9] and $O(mn \log mn)$-time algorithms producing Euclidean Voronoi diagrams for $O(mn)$ line segments (e.g., [10]). Given this new diagram, (P1) is solvable in linear time, and thus can be solved in $O(mn \log mn)$ total time.

(P2) *Locate the polygon P inside the polygonal region Q so that the minimum Euclidean distance between any point on P and any point on Q is maximized. In this problem, P can be rotated and translated.*

Chew and Kedem [8] study the related problem of finding the largest similar copy of P inside Q using translation and rotation. However, if the convex polygon P is thin, it is more natural to place P as in (P2). By considering the dynamic version of the Voronoi diagram in (P1), we show that problem (P2) can be solved in $O(m^4 n \lambda_{16}(mn) \log mn)$ time, where $\lambda_{16}(mn)$ is the maximum length of Davenport-Schinzel sequences of order 16 (see [2], [5], [15]). The currently best bound for $\lambda_{16}(N)$ is $O(N 2^{O(\alpha(N)^7)})$ [2], where $\alpha(N)$ is the functional inverse of Ackermann's function. $\lambda_{16}(N)$ is almost linear in N.

(P3) *Locate k copies of P inside Q so that the copies are horizontal translates of each other, the i-th copy being the horizontal translate of the $(i-1)$-th copy by h, and the minimum distance between any point on any copy of P and any point of Q is maximized, where h is a variable greater than or equal to a given constant $h_0 > 0$ so that the copies of P at regular interval h between the corresponding pair of reference points do not intersect one another.*

For a fixed h, this problem can be solved by computing the intersection of k copies of the polygonal region Q at regular intervals of h, and then considering (P1) for this intersection. For this problem with h as a variable, we consider a dynamic Voronoi diagram for k moving copies of Q horizontally at regular intervals of h, with h ranging from h_0 to ∞, and analyze the combinatorial complexity of this diagram. We show that, for $k = 2$ and $k \geq 3$, problem (P3) can be solved in $O(m^2 n^2 \log mn)$ and $O(k^6 m^3 n^3 \log kmn)$ time, respectively.

2. Maximin Placement of P inside Q by Translation

In this section, we explain our solution to problem (P1) of the maximin location of the convex polygon P of m vertices inside the polygonal region Q of n vertices, using translation only. This kind of maximin placement problem is often solved by using Voronoi diagrams for appropriately defined distance functions. In fact, if P is a point, this problem can be easily solved by first constructing in Q the Voronoi diagram for the edges of Q based on the Euclidean distance, and then finding a Voronoi point at which the distance to the

nearest edge is maximum. Our approach introduces a new Voronoi diagram suitable for this problem, which will also be used in solving (P2) and (P3).

We suppose the convex polygon P in the plane is given together with a reference point p inside P. For a point u, we denote by $P(u)$ the polygon obtained by translating P so that the reference point p coincides with u.

The *feasible region* of P inside Q [9], [14] is defined to be a set of points u inside Q such that $P(u)$ is contained in Q. The feasible region may consist of several connected components, and each component is a polygon. The boundary polygon(s) of the feasible region is denoted by $F(P, Q)$. It is shown by Fortune [9], Leven, Sharir [14] that the combinatorial complexity (the number of edges and vertices in this case) of the feasible region is $O(mn)$, and that it can be computed in $O(mn \log mn)$ time by constructing the Voronoi diagram of Q with respect to the convex distance function concerning P.

We now define a new Voronoi diagram variant. In considering the Voronoi diagram, a line segment is considered to consist of two endpoints and an open line segment. Vertices and (open) edges of a polygon are called *faces* of the polygon, and so the Voronoi diagram is for faces of P and Q. Faces of P and Q are especially called *P-faces* and *Q-faces*, respectively.

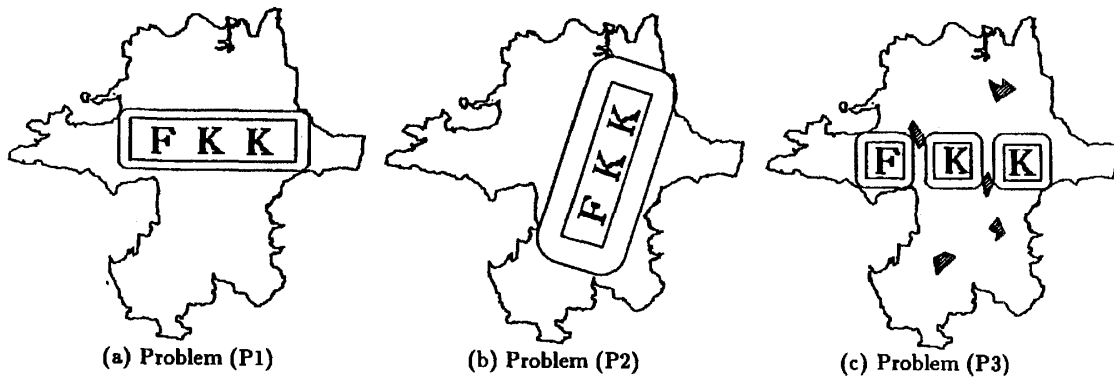(a) Problem (P1)   (b) Problem (P2)   (c) Problem (P3)

Figure 1. Placing a regional name on a map

Let r be a P-face and s a Q-face. For a point u, define $d_b(u; r, s)$ to be the distance between P-face r of $P(u)$ and Q-face s. Here, the distance between two points is just the ordinary Euclidean distance, and the distance between a point v and an open line segment w is defined to be the length of the perpendicular line segment from the point v to the line containing w if the perpendicular line intersect w, and $+\infty$ otherwise. The distance between two non-intersecting open line segments is generally considered to be $+\infty$, but, when they are parallel and there exists a perpendicular line segment connecting two points on the open line segments, the distance is defined to be the length of the perpendicular line segment.

Define the boundary distance $d_b(P(u), Q)$ by

$$d_b(P(u), Q) = \min\{ d_b(u; r, s) \mid r : P\text{-face}, \ s : Q\text{-face} \}$$

Using this notation, Problem (P1) may be restated as follows:

(P1) $\max\{ d_b(P(u), Q) \mid u \text{ in } F(P, Q) \}$

The Voronoi region $V(r, s)$ of P-face r and Q-face s is defined as

$$V(r, s) = \bigcap \{ u \text{ in } F(P, Q) \mid d_b(u; r, s) < d_b(u; r', s'),$$
$$r'(\neq r) : \forall P\text{-face}, \ s'(\neq s) : \forall Q\text{-face} \}$$

The planar skeleton V formed by the boundaries of $V(r, s)$ (r: P-face, s: Q-face) is called the *P-Euclidean Voronoi diagram*. The common boundary of two Voronoi regions $V(r, s)$ and $V(r', s')$ is called a Voronoi edge. We call these two pairs (r, s) and (r', s') of a P-face and a Q-face the *supporting pairs* of the Voronoi edge, and say that the Voronoi edge is supported by (r, s) and (r', s'). The common boundary of three Voronoi regions is called a Voronoi vertex, as usual. This diagram plays an important role in the following sections.

In defining the Voronoi region above, if there is a parallel pair of an edge r of P and an edge s of Q such that there is a perpendicular line segment connecting two points on the two edges, the distance between two open edges r and s is equivalent to the distance from an endpoint of one of r and s to the other edge (suppose the endpoint is an endpoint r' of r and the other edge is s). Then, $V(r, s)$ and $V(r', s)$ are both empty, which is not good. To avoid this, we regard the distance between r and s is smaller than the distance from the endpoint r' of edge r to edge s. When there is an edge of P and two concave vertices of Q such that the line connecting two concave vertices is parallel to the edge of P, we break a tie by regarding a vertex with lexicographically smaller coordinate is closer to the edge. We call these two special cases *parallel degenerate cases*, and handle them in such a special way.

The diagram is a subdivision of the interior of the feasible region $F(P, Q)$. It has the following basic properties.

Lemma 2.1. Considering four cases where r and s are a vertex/edge of P and Q, respectively, we have the following.
(a) $V(r, s) = \emptyset$ for an edge r of P and an edge s of Q unless r and s form a parallel degenerate case.
(b) For an edge s of Q, there is at most one P-face r with $V(r, s) \neq \emptyset$, and then r is a vertex of P unless r and s form a parallel degenerate case.
(c) For a convex vertex s of Q, $V(r, s) = \emptyset$ for any P-face r.
(d) For a concave vertex s of Q, there may be multiple P-faces r with $V(r, s) \neq \emptyset$. $\square$

Lemma 2.2. There are $O(mn)$ Voronoi regions, edges and vertices. $\blacksquare$

For a point u on a boundary edge w of the feasible region $F(P, Q)$, $P(u)$ has a contact with the boundary of Q. Such contacts can be classified into four types as follows.
(a) a contact between a vertex of $P(u)$ and a vertex of Q: u is then a vertex of $F(P, Q)$.
(b) a contact between a vertex of P and an edge of Q: w is then parallel to the edge of Q.
(c) a contact between an edge of P and a concave vertex of Q: w is then parallel to the edge of P.
(d) a contact between an edge in P and an edge in Q: this is a parallel degenerate case, and should be handled differently as mentioned above.

We now show a main lemma for the P-Euclidean diagram. Define $d(F, u)$ to be the minimum Euclidean distance between any point on the boundary of $F = F(P, Q)$ and u.

Lemma 2.3. For a point u inside the feasible region, $d_b(P(u), Q) = d(F, u)$.

Proof: We first show $d_b(P(u), Q) \leq d(F, u)$. Suppose that $d(F, u)$ is attained by u and a point u' on the boundary of $F(P, Q)$. Let v' be the point of $P(u')$ in contact with Q, and let v the point of $P(u)$ corresponding to v' (see Figure 2).

Since $P(u')$ is a translate of $P(u)$, clearly $d_2(v', v) = d_2(u', u)$, where d_2 denotes the Euclidean distance between two points. Then

$$d_b(P(u), Q) \leq d_2(v', v) = d_2(u', u) = d(F, u).$$

To show that $d_b(P(u), Q) \geq d(F, u)$, we let v' and v be the points of Q and $P(u)$ determining $d_b(P(u), Q)$, respectively. Let $P(u')$ be the translate of $P(u)$ in contact with Q at v' (again, see Figure 2). The point u' is thus either on the boundary of F, or outside F.

But, in either case,

$$d_{\mathrm{b}}(P(u), Q) = d_2(v', v) = d_2(u', u) \geq d(F, u),$$

and hence $d_{\mathrm{b}}(P(u), Q) = d(F, u)$. □

Lemma 2.3 implies that the Euclidean Voronoi diagram for the line segments of $F(P, Q)$ and the P-Euclidean diagram are identical over the feasible region $F(P, Q)$. The Euclidean Voronoi diagram for $O(mn)$ lines segments can be constructed in $O(mn \log mn)$ time (e.g., see Fortune [10]). Therefore, we have the following lemma.

Lemma 2.4. The P-Euclidean diagram can be constructed in $O(mn \log mn)$ time. □

Problem (P1) is now reduced to finding a point u inside the feasible region $F(P, Q)$ such that the minimum distance between u and any point on $F(P, Q)$ is maximized. This is equivalent to finding the largest enclosed circle inside $F(P, Q)$. Given the P-Euclidean diagram, the largest enclosed circle in it can be found in linear time. Thus, we have the following theorem:

Theorem 2.1. Problem (P1) can be solved in $O(mn \log mn)$ time by using the P-Euclidean diagram. □
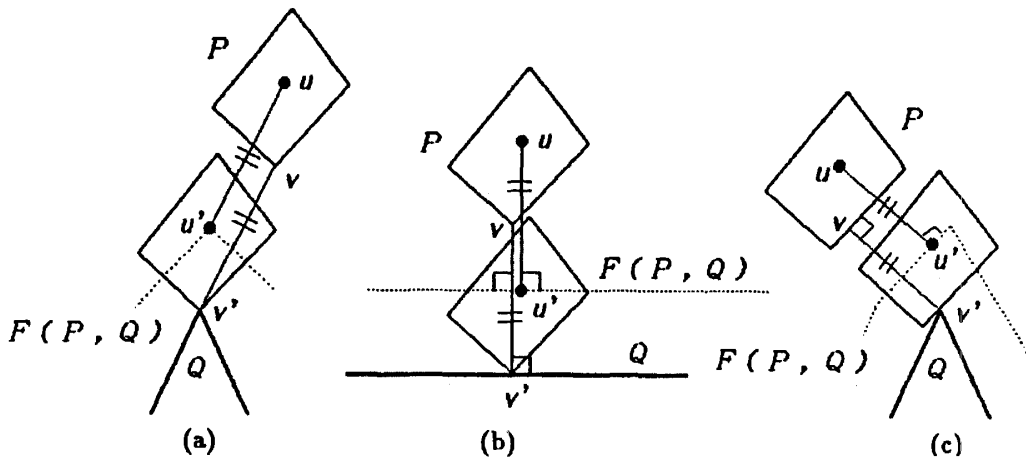

Figure 2. Proof of Lemma 2.3

3. Maximin Placement of P inside Q by Translation and Rotation

As shown in the previous section, without rotation, the P-Euclidean diagram directly gives a solution to problem (P1). However, when the orientation of P changes, the feasible region $F(P, Q)$ itself changes, and the P-Euclidean diagram also changes dynamically. In order to solve problem (P2) by using a Voronoi diagram, we have to compute the dynamic P-Euclidean diagram. This is similar to the situation encountered in the problem of finding the largest similar copy of P inside Q, considered by Chew and Kedem [8]. We investigate the combinatorial complexity of the dynamic P-Euclidean diagram based on their approach. However, the P-Euclidean diagram is much more complicated than the diagram defined by the convex distance function considered in [8], and several new ideas are necessary to get a result for our problem.

To represent the orientation of P, for an angle θ ($0 \leq \theta \leq 2\pi$), we consider the polygon $\widetilde{P}(\theta)$ obtained by rotating P by θ around the reference point p. For a point u in the feasible region $F(\widetilde{P}(\theta), Q)$, the polygon obtained by translating $\widetilde{P}(\theta)$ so that p coincides with u is denoted by $\widetilde{P}(\theta, u)$.

We will first investigate some properties of the static P-Euclidean diagram which are useful in analyzing the dynamic diagram, and then consider the combinatorial complexity of the dynamic diagram in two steps.

3.1. Useful properties of the P-Euclidean diagram

In this subsection, we classify Voronoi edges of the P-Euclidean diagram into two types.

For a Voronoi edge e of the P-Euclidean diagram, let (r, s) and (r', s') be the two supporting pairs of the edge, where r, r' are P-faces and s, s' are Q-faces. These pairs are distinct, that is, $(r, s) \neq (r', s')$, but either $r = r'$ or $s = s'$ may hold. If $r = r'$, these two are a vertex of P, and, if $s = s'$, these two are a concave vertex of Q. We call the Voronoi edge *proper* if $r = r'$ or $s = s'$ holds. Also, in this case, the pair (s, s') of Q-faces is called a *proper Q-pair*. For a concave vertex s of Q, (s, s) is a proper Q-pair. A Voronoi vertex incident to a proper Voronoi edge is defined to be *proper*. Voronoi edges and vertices which are not proper are defined to be *improper*.

Let v be a Voronoi vertex incident to e. There is another supporting pair (r'', s'') besides the two pairs at the Voronoi vertex.

Lemma 3.1. Suppose v is an improper Voronoi vertex. Regard e as a directed edge toward v and consider a directed line containing e and having the same direction. Suppose r and r' are in the left and right sides of the directed line, respectively. Then, r'' is distinct from r and r', and is contained in the list of P-faces from r to r' in clockwise order (in the list vertices and edges of P appear alternatively).

Proof: Since v is improper, r'' is distinct from r and r'. Hence, r'' is either a member of P-faces from r to r' in clockwise order or a member of P-faces from r' to r in clockwise order. But the latter case cannot occur, which may be naturally seen from the fact that the P-Euclidean diagram is identical with the ordinary Euclidean Voronoi diagram for the feasible region (Lemma 2.3).   □

Lemma 3.2. Suppose all the improper Voronoi edges are cut at all the proper Voronoi vertices (i.e., improper Voronoi edges incident to the same proper Voronoi vertex are made to be non-adjacent to each other), and then are decomposed into the connected components. Then, the number of improper Voronoi edges and vertices in each connected component is $O(m)$.

Proof: We consider a component containing an improper Voronoi vertex v. In the original diagram, the diagram is a tree, and we transform this tree to a directed tree rooted at v. For each directed Voronoi edge whose supporting pairs of P-faces are r and r' such that, with respect to this directed edge, r is in the left side and r' is in the right side, we associate a list of P-faces from r to r' in clockwise consecutive order. In the list, vertices and edges of P appear alternatively, and we consider that r and r' themselves are not contained in the list.

We walk on the tree from v by a depth first search. In this walk, the search backtracks when it hits a proper Voronoi vertex. Then the number of vertices visited during this walk gives a bound on the number of improper Voronoi vertices in the connected component containing v. We will show that this number is $O(m)$.

We mark a P-face each time the search proceeds successfully. Initially only P-faces supporting the Voronoi vertex v are marked. Let v' be the current Voronoi vertex, and suppose that an adjacent Voronoi vertex v'' to v' is next visited along a Voronoi edge supported by the pairs (r, s) and (r', s'). If v'' is a vertex of the feasible region or a proper Voronoi vertex, the search backtracks.

Otherwise, as in Lemma 3.1, a new supporting pair (r'', s'') appears at v'' such that r'' is distinct from r and r', and r'' is in the list of P-faces associated with the directed Voronoi edge. r'' is then marked, and the depth first search proceeds. Since the search backtracks at a proper Voronoi vertex, and from Lemma 3.1, only P-faces in the list associated with

the directed Voronoi edge are marked once while descendants of v'' are visited, and these marked vertices will not be marked after the search backtracks over v''.

Thus each P-face is marked only once during the whole search and we mark a P-face each time the search proceeds forward. Therefore, the search proceeds successfully at most $2m$ times ($2m$ is the total number of P-faces), and it backtracks at most the same number of times. Therefore, this depth first search visits at most $4m$ vertices. □

This lemma implies that, if the number of topological changes of proper Voronoi vertices may be estimated, the total number of topological changes may be evaluated. We will first evaluate topological changes related to proper Voronoi edges, and then prove this observation later.

3.2. Topological changes of proper Voronoi vertices

We now consider the dynamic $\widetilde{P}(\theta)$-Euclidean diagram when θ moves from 0 to 2π. First, we consider the number of proper Q-pairs, where a trivial bound is $O(n^2)$.

Lemma 3.3. The number of proper Q-pairs is $O(n)$.

Proof: The number of proper Q-pairs which are pairs of the same concave vertex of Q is trivially $O(n)$. Let e be the Voronoi edge corresponding to a proper Q-pair of distinct Q-faces s and s'. This edge is supported by (r, s) and (r, s') for a vertex r of P. A circle centered at r of $\widetilde{P}(\theta, u)$ with radius $d_b(\widetilde{P}(\theta, u), Q)$ for a point u on e is contained in Q and touches Q at s and s'. This implies that, in the Euclidean Voronoi diagram for n line segments of Q, there is a Voronoi edge equidistant from s and s'. Since there are $O(n)$ Voronoi edges in the Euclidean Voronoi diagram, the lemma follows. □

Now, fix a proper Q-pair (s, s') of Q-faces. If $s \neq s'$, fix a vertex $r = r'$ of P; otherwise, fix a pair of a vertex r and its incident edge r' of P. For this, there may be a proper Voronoi edge supported by two supporting pairs (r, s) and (r', s') at some θ. To evaluate the number of topological changes of proper Voronoi vertices incident to this proper Voronoi edge, we classify proper Voronoi edges into four types:

(e1) proper edges supported by two edges s, s' of Q ($r(= r')$ is a vertex of P);

(e2) proper edges with a concave vertex $s = s'$ of Q (r is a vertex of P and r' is its incident edge), or proper edges supported by a concave vertex s and its incident edge s' of Q ($r(= r')$ is a vertex of P);

(e3) proper edges supported by a concave vertex s and an edge s' of Q which are not incident to each other ($r(= r')$ is a vertex of P);

(e4) proper edges supported by two concave vertices s, s' of Q ($r(= r')$ is a vertex of P).

Let e be a proper edge of type (e1) (see Figure 3). For a pair c of a P-face and a Q-face, define a function $d_c(\theta)$ to be the minimum δ such that, for some point u which is on the line containing e and in $F(\widetilde{P}(\theta), Q)$, the boundary distance $d_b(\widetilde{P}(\theta, u), Q) = \delta$ is attained by (r, s), (r, s') and the pair c. If there is no such a point u, $d_c(\theta)$ is set to $+\infty$. In Figure 3, the pair c are an edge of $\widetilde{P}(\theta, u)$ and a concave vertex of Q. Let $\mathcal{C}$ be a set of all pairs c of a P-face and a Q-face, and define a function $d_{\mathcal{C}}(\theta)$ by

$$d_{\mathcal{C}}(\theta) = \min_{c \in \mathcal{C}} d_c(\theta).$$

Then, if d_c for the pair c attains the minimum in $d_{\mathcal{C}}$, this pair, together with (r, s), (r, s'), determines a proper Voronoi vertex. Minima change as θ varies. $d_{\mathcal{C}}$ can be computed from the lower envelope of $d_c(\theta)$ ($c \in \mathcal{C}$), for which results on Davenport-Schinzel sequences can be used.

We can define $d_c(\theta)$ for edges of other types similarly. General functional forms of $d_c(\theta)$ are given as follows, where there are four cases for types of contact pairs c but the case for edge-edge contact pairs is skipped (all Greek letters with subscripts are constants).

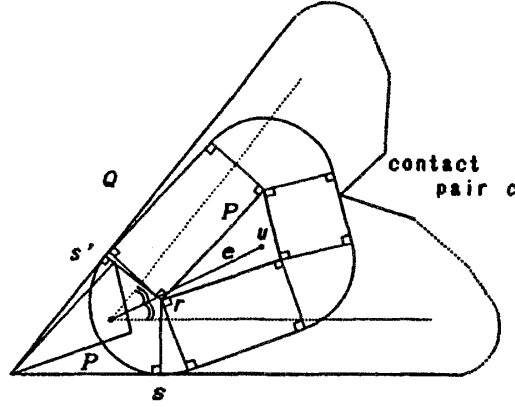For a Voronoi edge e of type (e1), $d_c(\theta)$ may be expressed as follows.

Figure 3. A proper Voronoi edge e of type
(e1) supported by (r, s) and (r, s') for a ver-
tex r of P and edges s, s' of Q

(a) c is a pair of a vertex of P and an edge of Q:

$$d_c(\theta) = \alpha_1 \cos(\theta + \alpha_2) + \alpha_3$$

(b) c is a pair of an edge of P and a vertex of Q:

$$d_c(\theta) = \frac{\beta_1 \cos(\theta + \beta_2) + \beta_3}{\beta_4 \cos(\theta + \beta_5) + \beta_6}$$

(c) c is a pair of a vertex of P and a vertex of Q:

$$(d_c(\theta))^2 + (\gamma_1 \cos(\theta + \gamma_2) + \gamma_3)d_c(\theta)$$
$$+ \gamma_4 \cos(\theta + \gamma_5) + \gamma_6 = 0$$

For a proper edge of type (e2), the same expressions are obtained for the cases (a) and
(b), and the following simplified expression is obtained for the case (c).
(c) c is a pair of a vertex of P and a vertex of Q:

$$d_c(\theta) = \frac{\gamma_1 \cos(\theta + \gamma_2) + \gamma_3}{\gamma_4 \cos(\theta + \gamma_5) + \gamma_6}$$

Next, we consider a proper Voronoi edge of type (e3). Here, we consider a variable x
defined by

$$d_c(\theta) = x^2 + 1, \qquad x \geq 0$$

and give expressions for x.
(a) c is a pair of a vertex of P and an edge of Q:

$$x^2 + \alpha_1 x + \alpha_2 \cos(\theta + \alpha_3) + \alpha_4 = 0$$

(b) c is a pair of an edge of P and a vertex of Q:

$$(\cos(\theta + \beta_1) + \beta_2)x^2 + \beta_3 x \sin(\theta + \beta_1)$$
$$+ \beta_4 \cos(\theta + \beta_5) + \beta_6 = 0$$

(c) c is a pair of a vertex of P and a vertex of Q:

$$(\cos(\theta + \gamma_1) + \gamma_2)x^2 + (\gamma_3 \cos(\theta + \gamma_4) + \gamma_5)x$$
$$+ \gamma_6 \cos(\theta + \gamma_7) + \gamma_8 = 0$$

Finally, we consider a proper Voronoi edge of type (e4). In this case we consider a variable x defined by
$$d_c(\theta) = \sqrt{x^2 + 1}, \qquad x \geq 0$$
and give expressions for x.

(a) c is a pair of a vertex of P and an edge of Q:

$$x^2 + (\alpha_1 \cos(\theta + \alpha_2) + \alpha_3)x$$
$$+ \alpha_4 \cos^2(\theta + \alpha_2) + \alpha_5 \cos(\theta + \alpha_2) + \alpha_6 = 0$$

(b) c is a pair of an edge of P and a vertex of Q:

$$x^2 \sin^2(\theta + \beta_1) + \cos(\theta + \beta_1)(\beta_2 \cos(\theta + \beta_3) + \beta_4)x$$
$$+ \beta_5 \cos^2(\theta + \beta_3) + \beta_6 \cos(\theta + \beta_3) + \beta_7 = 0$$

(c) c is a pair of a vertex of P and a vertex of Q:

$$(\cos(\theta + \gamma_1) + \gamma_2)x + \gamma_3 \cos(\theta + \gamma_4) + \gamma_5 = 0$$

Lemma 3.4. For a proper edge of types (e1), (e2), (e3) and (e4), any pair of functions d_c ($c \in C$) intersect at most 4, 4, 8 and 16 times, respectively. Hence, the combinatorial complexity of d_C is $O(\lambda_{16}(mn))$.

Proof: The number of intersections can be calculated directly. Then the lemma follows from the theory of Davenport-Schinzel sequences. $\square$

Lemma 3.5. The number of topological changes of proper Voronoi vertices is $O(mn \lambda_{16}(mn))$ in total.

Proof: From Lemma 3.3, there are $O(n)$ proper Q-pairs. With each proper Q-pair, we may pair each vertex of P, or for each proper Q-pair of the same concave vertex we may associate a pair of a vertex and its incident edge of P. Since the number of vertices of P is m, we may consider $O(mn)$ pairs, and then applying Lemma 3.4 we have this lemma. $\square$

3.3. Topological changes of improper Voronoi vertices

Now, we have to estimate the number of topological changes of improper Voronoi vertices. For this purpose, we may use Lemma 3.2, which states that every connected component of all the improper Voronoi edges cut at proper Voronoi vertices consists of $O(m)$ improper Voronoi vertices and edges. In each component, the number of supporting pairs is $O(m)$ since there are $O(m)$ Voronoi edges. We will consider how the set of supporting pairs in each of the connected components changes and how such changes affect improper Voronoi vertices topologically.

For an improper Voronoi edge, any supporting pair is a pair of a vertex of P and an edge of Q, or that of an edge of P and a concave vertex of Q except the parallel degenerate case.

The set of supporting pairs in the connected component may be updated if

(a) there is a topological change of a proper Voronoi vertex incident to the component, or

(b) there occurs a parallel degenerate case, or

(c) the connected component of improper edges is divided into two, or two connected components of improper edges are merged into one according as the connected component of the feasible region is divided into two, or two connected components of the feasible region are merged into one.

In the cases (a) and (b), the set of supporting pairs is updated by deleting a few pairs and/or inserting a few new pairs (in ordinary cases a pair is deleted and a new pair is inserted). In the case (c), the set of supporting pairs is greatly changed.

About the case (a), we have evaluated the number of such changes in Lemma 3.5. For the case (b), recall that the parallel degenerate cases are such that

(b1) there are a parallel pair of an edge r of P and an edge s of Q, or

(b2) there are an edge of P and two concave vertices of Q such that the line connecting the two concave vertices is parallel to the edge of P.

The case (b1) occurs $O(mn)$ times, and the case (b2) occurs $O(mn^2)$ times, and we thus have the following.

Lemma 3.6. The number of times a parallel degenerate case occurs is $O(mn^2)$. □

For the case (c), we have to evaluate the number of times the connected components of the feasible region change (as mentioned in section 2, the feasible region may consist of several connected components and these components dynamically change as θ varies). Considering when the connected components of the feasible region change, we have the following.

Lemma 3.7. When two connected components of the feasible region are merged into one, or a connected component is divided into two at $\theta = \theta'$, at the junction point u, $\widetilde{P}(\theta', u)$ is in Q and touches the boundary of Q at an antipodal pair of P-faces.

Proof: When a convex polygon is supported at two parts which are not antipodal to each other, the polygon may be moved in at most one direction along any line by translation. At $\theta = \theta'$, $\widetilde{P}(\theta', u)$ can move in both directions along some line, and so the lemma follows. □

Lemma 3.8. The number of changes of the connected components of the feasible region is $O(mn^2)$.

Proof: From Lemma 3.7, there may be a change for an antipodal pair of P-faces and a pair of Q-faces. Since the number of antipodal pairs of P-faces is $O(m)$ and the number of pairs of Q-faces is $O(n^2)$, the lemma follows. □

Thus we have evaluated how many times the set of supporting elements in each connected component is updated in total.

There can be topological changes in the component even if the set of supporting pairs is not updated. A tuple of four supporting pairs in the set can determine a degenerate Voronoi vertex of degree four, and so will be called a *candidate tuple*. For each candidate tuple of four supporting pairs, there are at most a constant number of topological changes determined by the tuple in total. For a set of $O(m)$ supporting pairs, there are $O(\binom{O(m)}{4}) = O(m^4)$ tuples.

Let us evaluate the number of distinct candidate tuples which appears in some set of supporting pairs in a component in the whole dynamic diagram.

Initially, there are $O(mn)$ connected components, and we may count $O(mn \cdot m^4)$ candidate tuples.

For the cases (a) and (b), when the set of supporting pairs is updated, only a few supporting pairs are deleted and a few possibly new supporting pairs are inserted to the set. The other $O(m)$ supporting pairs remain unchanged. Since there are a constant number of new supporting pairs, the number of new candidate tuples of four supporting pairs in the component is just $O(m^3)$, not $\Theta(m^4)$. Hence, we may just count $O(m^3)$ candidate tuples each time the case (a) or (b) occurs.

For each of the cases of (c), since the set of supporting pairs is updated much, we may directly count $O(m^4)$ candidate tuples for a set of $O(m)$ supporting pairs obtained by merging/dividing the connected component(s).

Thus, the total number of distinct candidate tuples in the whole dynamic diagram is bounded by

$$O(mn \cdot m^4 + (mn\lambda_{16}(mn) + mn^2)m^3 + mn^2 \cdot m^4)$$
$$= O(m^4 n\lambda_{16}(mn)).$$

As noted above, there may be a constant number of topological changes for each tuple when θ changes from 0 to 2π, we obtain the following.

Lemma 3.9. The number of topological changes of improper Voronoi vertices is $O(m^4 n \lambda_{16}(mn))$. □

3.4. Combinatorial complexity of the dynamic diagram

Regarding the original plane as the (x, y)-plane, the dynamic diagram is a diagram in the three-dimensional (x, y, θ)-space. Combining Lemmas 3.5 and 3.9, the combinatorial complexity of this dynamic diagram is given as follows.

Theorem 3.1. The combinatorial complexity of the $\widetilde{P}(\theta)$-Euclidean diagram in the (x, y, θ)-space with $0 \le \theta \le 2\pi$ is $O(m^4 n \lambda_{16}(mn))$. □

Algorithmically, we may first maintain the feasible region dynamically. As θ changes, the feasible region $F(\widetilde{P}(\theta), Q)$ of $\widetilde{P}(\theta)$ inside Q changes accordingly, and further a connected component of the feasible region may be divided into two connected components, and two disjoint connected components may be united to form a connected component. In computing the dynamic P-Euclidean diagram, we compute all the changes (especially, union of two connected components) of the feasible region in advance (cf. Lemma 3.8).

We can then apply the plane (strictly, space) sweep method using a heap, after computing the changes of the feasible region, and have the following theorem.

Theorem 3.2. The dynamic $\widetilde{P}(\theta)$-Euclidean diagram in the (x, y, θ) space with $0 \le \theta \le 2\pi$ can be constructed in $O(m^4 n \lambda_{16}(mn) \log mn)$ time, and problem (P2) can be solved in $O(m^4 n \lambda_{16}(mn) \log mn)$ time.

Proof: The time complexity follows from Theorem 3.1 and a fact that each operation for the heap requires $O(\log mn)$ time.

The problem (P2) for fixed θ (i.e., problem (P1)) can be solved by finding a Voronoi vertex in $\widetilde{P}(\theta)$-Euclidean diagram for this θ which is a center of the largest enclosed circle. Hence, for varying θ in the plane (space) sweep method, we maintain for each Voronoi vertex its starting time θ_{start} and, when it finishes at time θ_{finish}, we compute a time between θ_{start} and θ_{finish} such that the maximum enclosed circle centered at the Voronoi vertex is maximized. Then, by keeping track of the maximum value among these maximums for all Voronoi vertices, the problem (P2) can be solved, which can be performed within the time bound for the construction of the dynamic Voronoi diagram. □

4. Problem (P3)

Problem (P3) is originally stated in such a way that the k copies of P are located in Q at regular intervals. But, instead of considering how to place copies of P, for a fixed h, we may solve this problem by computing the intersection of k copies of the polygonal region Q at regular intervals of a fixed h, and then considering problem (P1) for the intersection.

The intersection of k copies of Q is of size $O(k^2 n^2)$ in the worst case, and can be computed in time linear to this size. For this intersection it takes $O(mk^2 n^2 \log mkn)$ time to solve problem (P1) (Theorem 2.1), and hence problem (P3) for a fixed h can be solved in this time.

In ordinary cases, k and m are much smaller than n, and are regarded as constants. Considering k and m as constants, the above time complexity is $O(n^2 \log n)$. We can solve problem (P3) for a fixed interval in $O(n^2)$ time by directly applying the plane-sweep paradigm where each of $O(n)$ one-dimensional subproblems can be solved in $O(n \log n)$ time individually, but each can be solved in $O(n)$ time by solving them as a series of consecutive problems. However, this $O(n^2)$-time algorithm is rather complicated and it has almost no connection for the discussion below, we omit its details.

To solve the general problem (P3), where h is a variable, by this approach, we have to compute the dynamic P-Euclidean diagram for k horizontally moving copies of Q. Before analyzing the combinatorial complexity of this dynamic diagram, we consider the most

canonical problem of this kind, that is, the problem of analyzing the combinatorial complexity of the Voronoi diagram for k rigidly moving sets of n points, which was first considered in Tokuyama [16]. The general problem is treated in [12], [13]. The following approach leads to a worse bound than the best known bound in [11] for such point set case, but is applicable to more general cases like our problem.

We consider the case of $k \geq 4$. There are k sets of n points in the (x, y)-plane at time $t = 0$, and as time passes each set moves smoothly according to the specified functions. Consider the three-dimensional (x, y, t)-space with $t \geq 0$. For each set, consider a spatial subdivision such that its intersection with the plane $t = t'$ gives a Voronoi diagram at time $t = t'$. Since each set is rigidly moving, the diagram at the intersection is congruent with the initial diagram.

Consider a subdivision obtained by overlaying these k spatial subdivisions. It is then easy to show the following.

Lemma 4.1. The combinatorial complexity (the number of regions, faces, edges and vertices) of the overlaid subdivision is $O(k^3 n^3)$. $\square$

Now, take a region R in the overlaid subdivision, and consider the number of topological changes in the region. This region is the intersection of k Voronoi regions, one from each of the Voronoi diagrams for k sets. Let S_R be a set of k points associated with these k Voronoi regions. Any topological change inside R is performed by four points in S_R, since the closest point to any point inside R for any of k sets of points is contained in S_R. Then the same arguments in section 3.3 can be applied, and we can show that the topological changes inside regions of the three-dimensional subdivision is $O(k^6 n^3)$ in total.

There are also topological changes on faces of the subdivision. However, we have already shown that the number of topological changes on faces of the subdivision is $O(n^2 k^3 \log^* k)$. We thus see that the combinatorial complexity of the dynamic Voronoi diagram for k rigid sets of n points is $O(k^6 n^3)$ for $k \geq 4$. This bound is worse than the bound $O(n^2 k^2 \lambda_s(k))$ given in [11], but this technique can be applied to the problem (P3) as follows.

Now we consider the problem (P3). As described above, we will consider the dynamic P-Euclidean diagram for $\bigcap_{i=0}^{k-1}(Q + ih)$, where $Q + ih$ denotes the translate of Q to the right by ih.

Applying the above technique, we consider k spatial subdivisions for the P-Euclidean diagram of $Q + ih$ ($i = 0, \ldots, k - 1$), and overlay these k spatial subdivisions. The overlaid subdivision consists of $O(k^3 N^3)$ regions, edges and vertices similarly, where $N = mn$. Therefore, the number of topological changes inside regions of the overlaid subdivision is $O(k^6 N^3)$. The number of topological changes on faces of the overlaid subdivision is bounded by this number.

Especially, in the case of $k = 2$, the combinatorial complexity of the overlaid subdivision is just $O(N^2)$, and the combinatorial complexity of the whole dynamic Voronoi diagram is $O(N^2)$.

In the case of of $k = 3$, however, unlike the point set case, the number of intersections between edges when three polygonal regions Q, $Q + h$, $Q + 2h$ move linearly as h increases is $O(N^3)$. From this, it is seen that the number of topological changes of the whole dynamic diagram is $O(N^3)$ (in the point set case, this bound is $O(N^2)$).

Thus, the combinatorial complexity of the whole dynamic Voronoi diagram for parameter h ranging from h_0 to $+\infty$ is $O(N^2)$ for $k = 2$ and $O(k^6 N^3)$ for $k \geq 3$. Algorithmically, again applying a plane (space) sweep algorithm with a heap, we obtain the following.

Theorem 4.1. Problem (P3) can be solved in $O(m^2 n^2 \log mn)$ and $O(k^6 m^3 n^3 \log kmn)$ time for $k=2$ and $k \geq 3$, respectively, using $O(kmn)$ space. $\square$

In the above algorithm, we consider k linearly moving polygons. The technique is applicable to much complicated movements, and can be directly applied to the following problem.

(P4) *Locate k translates of P inside Q so that the translates are at regular intervals h for a given h arranged along a line with slope $\tan\theta$ so that the minimum distance between any point on any translate of P and any point of Q is maximized. Here, θ is a variable, while h is constant.*

For this problem, we have the following.

Theorem 4.2. Problem (P4) can be solved in $O(m^2n^2\log mn)$ and $O(k^6m^3n^3\log kmn)$ time for $k=2$ and $k \geq 3$, respectively, using $O(kmn)$ space. □

It is left open whether the technique in [11] giving an quadratic bound for a constant number of rigid point sets can be applied to the above problems.

5. Concluding Remarks

In this paper we have introduced a new Voronoi diagram, called the P-Euclidean diagram, and its dynamic versions to solve the maximin placement of a convex polygon P inside a general polygon Q. This problem has much connection with the problem of placing a regional name on a map and that of finding a high-clearance path in the collision avoidance problem. We have also investigated the dynamic Voronoi diagram for k rigidly moving sets of points. Our bounds on the combinatorial complexity of the dynamic Voronoi diagrams improves trivial bounds much. Improving the bounds here or proving tight lower bounds for them would be an interesting problem.

Acknowledgment

References

[1] P. K. Agarwal and M. Sharir: *Davenport-Schinzel Sequences and Their Geometric Applications* (Cambridge University Press, Cambridge, UK, 1995).

[2] P. K. Agarwal, M. Sharir and P. Shor: Sharp upper and lower bounds for the length of general Davenport-Schinzel sequences. *Journal of Combinatorial Theory, Series A*, **52** (1989) 228–274.

[3] H. Aonuma, H. Imai, K. Imai and T. Tokuyama: Maximin location of convex objects in a polygon and related Voronoi diagrams. *Proceedings of the 6th ACM Symposium on Computational Geometry* (Berkeley, June 1990) 225–234.

[4] H. Aonuma, H. Imai and Y. Kambayashi: A visual system of placing characters appropriately in multimedia map databases. In *Visual Database Systems* (Proceedings of the IFIP TC2/WG2.6 Working Conference on Visual Database Systems, North-Holland, 1989) 525–546.

[5] M. J. Atallah: Some dynamic computational geometry problems. *Computers and Mathematics with Applications*, **11** (1985) 1171–1181.

[6] F. Avnaim and J. D. Boissonnat: Polygon placement under translation and rotation. *RAIRO Informatique Théorique et Applications*, **23**-1 (1989) 5–28.

[7] B. Chazelle: The polygon containment problem. In F. P. Preparata, ed.: *Advances in Computing Research. Vol.1: Computational Geometry* (JAI Press, Greenwich, Connecticut, 1983) 1–33.

[8]   L. P. Chew and K. Kedem: A convex polygon among polygonal obstacles: placement and high-clearance motion. *Computational Geometry: Theory and Applications*, **3-2** (1993) 59–89.

[9]   S. Fortune: Fast algorithms for polygon containment. *Proceedings of the 12th ICALP*, Lecture Notes in Computer Science, **194** (Springer-Verlag, 1985) 189–198.

[10] S. Fortune: A sweepline algorithm for Voronoi diagrams. *Algorithmica*, **2** (1987) 153–174.

[11] D. P. Huttenlocher, K. Kedem and J. M. Kleinberg: On dynamic Voronoi diagrams and the minimum Hausdorff distance for point sets under Euclidean motion in the plane. *Proceedings of the 8th Annual ACM Symposium on Computational Geometry* (1992) 110–119.

[12] K. Imai: Voronoi diagram for moving points and its applications (in Japanese). *Transactions of the Japan Society for Industrial and Applied Mathematics*, **1-2** (1991) 127–134.

[13] K. Imai, S. Sumino and H. Imai: Minimax geometric fitting of two corresponding sets of points and dynamic furthest Voronoi diagrams. *IEICE Transactions on Information and Systems*, **E81-D-11** (November 1998) 1162–1171.

[14] D. Leven and M. Sharir: Planning a purely translational motion for a convex polygonal object in two dimensional space using generalized Voronoi diagrams. *Discrete and Computational Geometry*, **2** (1987) 255–270.

[15] E. Szemerédi: On a problem by Davenport and Schinzel. *Acta Arithmetica*, **25** (1974) 213–224.

[16] T. Tokuyama: Deformation of merged Voronoi diagrams with translation. *TRL Research Report TR-88-0049*, IBM Tokyo Research Laboratory, 1988.

Keiko Imai
Department of Information
   and System Engineering
Chuo University
Kasuga, Bunkyo-ku
Tokyo, 112-8551 Japan
E-mail: imai@ise.chuo-u.ac.jp