Compact Voronoi Diagrams for Moving Convex Polygons

Leonidas Guibas* Jack Snoeyink† Li Zhang‡

Abstract

We describe a kinetic data structure for maintaining a compact Voronoi-like diagram of convex polygons moving around in the plane. We use a compact diagram for the polygons, dual to the Voronoi, first presented in [MKS96]. A key feature of this diagram is that its size is only a function of the number of polygons and not of their complexity. We demonstrate a local certifying property of that diagram, akin to that of Delaunay triangulations of points. We then obtain a method for maintaining this diagram that is output-sensitive and costs $O(\log n)$ per update. Furthermore, we show that for a set of k polygons with a total of n vertices moving along bounded degree algebraic motions, this dual diagram, and thus their compact Voronoi diagram, changes combinatorially $\Omega(n^2)$ and $O(kn^2\beta(k)\beta(n))$ times, where $\beta(\cdot)$ is an extremely slowly growing function. This compact Voronoi diagram can be used for collision detection or retraction motion planning among the moving polygons.

1 Introduction

Voronoi diagrams are fundamental data structures in computational geometry and have been used in a wide variety of applications that require proximity information among geometric objects. When a Voronoi diagram is defined on objects that are not points, all features of these objects can contribute to the diagram's complexity. In this paper we will be concerned with the Voronoi diagram of k convex polygons in the plane with a total of n vertices. Even though such a diagram defines only k regions (one per object), its total geometric complexity is $\Theta(n)$ — as all polygon vertices can contribute to the linear or parabolic bisector segments defining the edges separating these regions. Many of the queries we may want to use such a diagram for, however, (such as reporting the object closest to a query point, or the closest pair among the given objects) refer only to the objects themselves and not their individual features. In 1996, McAllister, Kirkpatrick, and Snoeyink [MKS96] showed how to compute what they called a *compact* Voronoi diagram, which is a simplified partition of the plane of size $\Theta(k)$, but which can still be used to answer proximity queries (such as the above) about the objects efficiently.

Voronoi-based methods have been successfully used to address proximity queries in robotics applications, such as collision detection [LC91] or retraction motion planning [Lat91]. In the robotics setting the convex polygons represent obstacles to be avoided. When these obstacles move, we need to update their Voronoi diagram accordingly. A natural framework for studying this is the *kinetic data structures* (KDS) framework, introduced by Basch, Guibas, and Hershberger in [BGH97]. In the KDS setting one maintains a geometric structure under continuous motion of its defining elements through a set of certificates proving its correctness. An event queue is maintained for the failure times of these certificates and at each event the structure of interest, and its kinetic proof,

*Department of Computer Science, Stanford University, Stanford, CA 94305. guibas@cs.stanford.edu.
†Department of Computer Science, University of North Carolina, Chapel Hill, NC 27599. snoeyink@cs.unc.edu.
‡Department of Computer Science, Stanford University, Stanford, CA 94305. lizhang@cs.stanford.edu.

are appropriately updated. It turns out that maintaining kinetically the Voronoi diagram of moving points is easy [GMR92], as a set of local conditions (`InCircle` tests) certify the global correctness of the structure and local repairs are always possible.

In this paper we study the kinetic maintenance of the compact Voronoi diagram for disjoint moving convex polygons in the plane. Though this diagram contains much less detailed information than the full Voronoi diagram, it turns out that it still has enough structure that it can be certified through a set of topologically local certificates and thus maintained as the objects move. This kinetic diagram can, for example, be used to track the closest pair among the moving objects and therefore perform collision detection. Many collision detection algorithms for convex bodies rely on determining the closest pair of features between two objects as the basic building block [LC91, Mir97]. To avoid considering all $\binom{k}{2}$ pairs of objects, these algorithms invoke a so-called 'broad-phase' method to select which pairs of objects to test — usually a intersection test on bounding boxes for the objects. The compact Voronoi diagram elegantly solves the broad-phase problem and always provides us with a set of $O(k)$ pairs of objects (those whose regions are adjacent in the diagram) that is guaranteed to contain the closest pair. As another example, the diagram can be used to solve the retraction motion planning problem, with the help of an additional structure that maintains the closest pair of features between two moving convex chains.

We introduce the basic notations and definitions we need in Section 2. A key notion is that of *junction triangles*, which are dual to the degree-3 vertices of the Voronoi diagram. If we remove the junction triangles from the free space around the obstacles, the rest of the free space can be decomposed into a set of corridors, each between two of the convex objects. This structure and the certification of its correctness are introduced in Section 3. In Section 4 we study the number of changes to the compact Voronoi diagram when the defining polygons move pseudo-algebraically in the plane. Using lower envelope and other techniques, we can show that the number of changes to the diagram is roughly $O(kn^2)$. Finally in Section 5 we give applications to the collision detection and retraction motion planning.

2 Preliminaries

A distance function δ defined on points in $\mathbb{R}^2$ can be generalized to points sets S_1 and S_2 by setting $\delta(S_1, S_2) = \inf_{s_1 \in S_1, s_2 \in S_2} \delta(s_1, s_2)$; when S_1 contains only one point s, we can simply write $\delta(s, S_2)$. If S_1, S_2 are bounded closed sets, their distance can be realized by a pair of points (s_1, s_2) where $s_1 \in S_1$ and $s_2 \in S_2$. In what follows we will use δ to denote the usual Euclidean distance.

Consider now a set $\mathcal{P}$ of disjoint convex obstacles in the plane. Under the above distance function between points and points sets, we can define a Voronoi diagram for $\mathcal{P}$, called the *generalized Voronoi diagram* of $\mathcal{P}$, which is the partition of the free space in the plane according to the nearest obstacle. We assume that $\mathcal{P}$ is a set of k disjoint convex polygons with n vertices in total. The generalized Voronoi diagram of $\mathcal{P}$, denoted by $\mathcal{V}(\mathcal{P})$, has complexity $O(n)$ and can be built in time $O(n \log n)$. In [MKS96], a compact representation of $\mathcal{V}(\mathcal{P})$ is presented. The proposed compact Voronoi diagram has size $O(k)$ and can be computed in $O(k \log n)$ time, assuming each object is represented by the sorted list of its vertices in clockwise (or counterclockwise) order. Despite its compactness, this new diagram is as powerful as the generalized Voronoi diagram with regards to nearest-neighbor and other queries. Here, we will show how to maintain this compact Voronoi diagram when the convex obstacles are in motion.

¿From this point on, when we refer to "an obstacle," we mean a closed convex polygon. An *edge* on a polygon is an open line segment; a *feature* of a polygon is a vertex or an (open) edge; the *size* of a polygon is the number of vertices on it. Two features are *adjacent* if their closures intersect.

Three polygons are *collinear* if there is a line tangent to them simultaneously. Four polygons are *cocircular* if there is a circle tangent to them simultaneously. Normally in computational geometry we assume that objects are in general position, and specifically in our setting that no two polygon edges are parallel, no three objects are collinear, and no four objects are cocircular. However, when objects can move, it is no longer legitimate to make the above assumption. Actually, interesting events happen exactly at the time when such a degeneracy occurs. For moving objects, by general position we mean that the above events happen at distinct discrete times (never two at once).

For any point p outside a polygon P, there is a unique point on P realizing the distance $\delta(p, P)$. Equivalently, there is a unique circle that is centered at p and tangent to P. Let us denote this circle by $\omega(p, P)$. The radius of $\omega(p, P)$ clearly equals $\delta(p, P)$. The unique feature (edge or vertex) that contains q is called the *closest feature* to p. For two disjoint convex polygons P and Q, if they do not have parallel edges, there is a unique pair of points (p, q), where $p \in P$ and $q \in Q$, that realizes $\delta(P, Q)$. We denote by $o(P, Q)$ the middle point of the line segment pq. At the point $o(P, Q)$, we can place a minimum circle that is tangent to both P and Q.

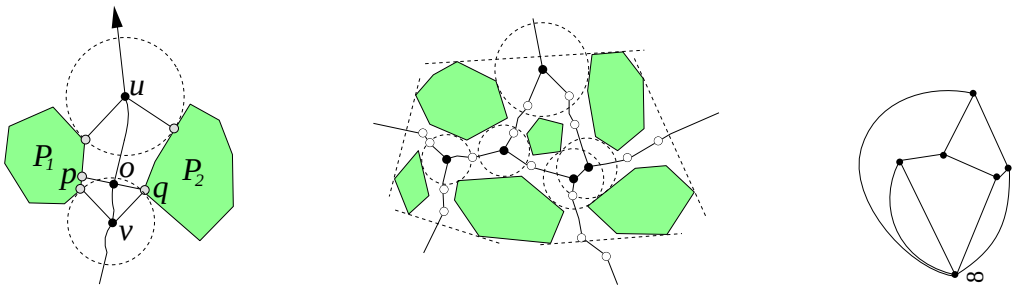
For the polygon set $\mathcal{P}$, denote the convex hull of the vertices of $\mathcal{P}$ by $\mathcal{C}(\mathcal{P})$. Let $\mathcal{F}(\mathcal{P}) = \mathcal{C}(\mathcal{P}) \setminus \mathcal{P}$ denote the free space outside the polygons but inside $\mathcal{C}(\mathcal{P})$. For polygons P_1 and P_2 in $\mathcal{P}$ and points $p \in P_1$ and $q \in P_2$, the edge pq is called a *free edge* if it does not intersect the interior of any polygon in $\mathcal{P}$. Two non-intersecting free edges pq and $p'q'$, where points $p, p' \in P_1$ and $q, q' \in P_2$, define a *corridor* with respect to $\mathcal{P}$ if and only if the region bounded by pq, $p'q'$, and the convex chains pp' on P_1 and qq' on P_2 contains no polygons of P. Sometimes we will talk about the *corridor of two polygons P_1 and P_2*, which is the corridor with respect to $\{P_1, P_2\}$ defined by their outer common tangents.

For two disjoint convex polygons P_1 and P_2, the bisector between them is defined to be the locus of points that are equidistant from them. It is well known that the bisector is an unbounded Jordan curve that consists of $O(|P_1|+|P_2|)$ line segments and parabolic arcs. For presentation convenience, we also add an orientation to the bisector. The oriented bisector $\pi(P_1, P_2)$ (abbreviated π_{12}) is the bisector with the orientation so that P_1 is to the left of π_{12}. Since the bisector is an oriented unbounded Jordan curve, we can define a linear ordering $\prec$ on the points on π_{12}. For two points $p, q \in \pi_{12}$, we say that $p \prec q$ if p is encountered before q when traveling on π_{12} consistently with the orientation of π_{12} (Figure 1 (a)). We can parameterize the bisector π_{12} as follows: for a point $p \in \pi_{12}$, if $p \prec o(P_1, P_2)$, then $\zeta_{12}(p) = -(\delta(p, P_1) - \delta(o, P_1))$; otherwise, $\zeta_{12}(p) = \delta(p, P_1) - \delta(o, P_1)$. When there is no degeneracy, the function ζ_{12} is an one-to-one and onto mapping from π_{12} to $\mathbb{R}$ ([MKS96]). Clearly, $\zeta_{12} = -\zeta_{21}$.

For three convex polygons P_1, P_2 and P_3 in general position, it is known that the bisectors π_{12} and π_{13} intersect at most twice. For an intersection v between π_{12} and π_{13}, the circle $\omega(v, P_1)$ is tangent to P_1, P_2 and P_3. We say that v is defined by the ordered triplet (P_1, P_2, P_3) if P_1, P_2 and P_3 are tangent to $\omega(v, P_1)$ in counterclockwise direction. Then there is at most one vertex defined by (P_1, P_2, P_3).

A point $p \in \pi_{12}$ is said to be *shaded* by P_3 if $\omega(p, P_1) \cap P_3 \neq \emptyset$. Let $S_{12,3}$ denote the set of the points on π_{12} shaded by P_3. Consider the set of parameter values represented by the shaded portion, $\{\zeta_{12}(p) \,|\, p \in S_{12,3}\}$, which we denote by $\tilde{S}_{12,3}$. Since bisectors π_{12} and π_{13} intersect at most twice, the shaded set $\tilde{S}_{12,3}$ must have the form $\emptyset$, $(-\infty, a]$, $[b, +\infty)$, $(-\infty, a] \cup [b, +\infty)$, $[a, b]$, or $(-\infty, +\infty)$, where a, b correspond to the parameter values of the Voronoi vertices defined by P_1, P_2 and P_3.

When $\tilde{S}_{12,3}$ has the form $(-\infty, a]$ or $(-\infty, a] \cup [b, +\infty)$, we say that π_{12} is *half-shaded* by P_3 at a. Notice that if neither π_{12} nor π_{21} is half-shaded by P_3, then $\tilde{S}_{12,3}$ must have the form $\emptyset$ or $[a, b]$ where $a < b$. The following fact is useful later in bounding the number of combinatorial changes.



(a) (b) (c)

Figure 1. (a) The oriented bisector π_{12} between P_1 and P_2. The pair (p, q) is the closest pair of points between P_1 and P_2. According to the ordering, $v \prec o \prec u$. (b) The generalized Voronoi diagram where the solid vertices are junction vertices and the hollow ones are interior vertices. Not all hollow vertices are shown. (c) The compact Voronoi diagram with only the junction vertices remaining. A point at ∞ is added to compactify the diagram.

Fact 2.1. *The shaded set $\tilde{S}_{12,3}$ is of the form $[a, b]$, where $a < b$, if and only if P_3 lies completely inside the corridor between P_1 and P_2.*

The bisector π_{12} divides the plane into two regions that contain P_1 and P_2, respectively. Let us denote $\tau_{P_1P_2}$(or simply τ_{12}) the region that contains P_1. Each point in τ_{12} is closer to P_1 than to P_2. For $P_i \in \mathcal{P}$, the Voronoi region $V(P_i)$ of P_i is then defined to be $\bigcap_{j \neq i} \tau_{ij}$ — the set of points that are closer to P_i than to any other polygon in $\mathcal{P}$. Each Voronoi region is connected and bounded by portions of bisectors. Therefore, the corresponding Voronoi diagram $\mathcal{V}(\mathcal{P})$ is a planar map (Figure 1 (b)). Two convex polygons are *adjacent* in $\mathcal{V}(\mathcal{P})$ if their Voronoi regions share a boundary.

Fact 2.2. *A point $p \in \pi_{12}$ is in $\mathcal{V}(\mathcal{P})$ if and only if $\zeta_{12}(p)$ is not in the interior of $\tilde{S}_{12,i}$, for any $i \neq 1, 2$.*

In $\mathcal{V}(\mathcal{P})$ there are two types of vertices, as illustrated in Figure 1 (b). Vertices of degree three are *junction vertices*, corresponding to the intersections between two bisectors. The remaining degree two vertices are *interior vertices*, lying along a single bisector where the closest feature pair changes. At each junction vertex, we can grow a circle that touches three polygons and is free of other polygons. This circle is called a *witness circle* for that junction vertex.

The junction vertices capture the topology of $\mathcal{V}(\mathcal{P})$. Furthermore, while the total number of vertices in $\mathcal{V}(\mathcal{P})$ can be $\Theta(n)$, the number of junction vertices is at most $2k - 4$, where k is the number of polygons in $\mathcal{P}$. The compact Voronoi diagram is based on these junction vertices, plus an additional imaginary vertex at ∞: we form an edge between two vertices if and only if there is a portion of a bisector between them with no junction vertex in between (Figure 1 (c)). If a portion of bisector extends to infinity, we connect the vertex to the node at ∞. We refer to this diagram as the *compact Voronoi diagram* of $\mathcal{P}$. In [MKS96], it is shown that the compact Voronoi diagram has many useful properties and that it can be computed in $O(k \log n)$ time. In the following, we study how to maintain the compact Voronoi diagram for moving polygons.

3 Maintaining the compact Voronoi diagram for moving obstacles

Maintaining the traditional Voronoi diagram of moving points in the plane, usually represented via its dual Delaunay triangulation, is straightforward [GMR92]. This is because a local condition, the 'empty circle' property for triangulation edges with pairs of adjacent triangles [PS90], implies the

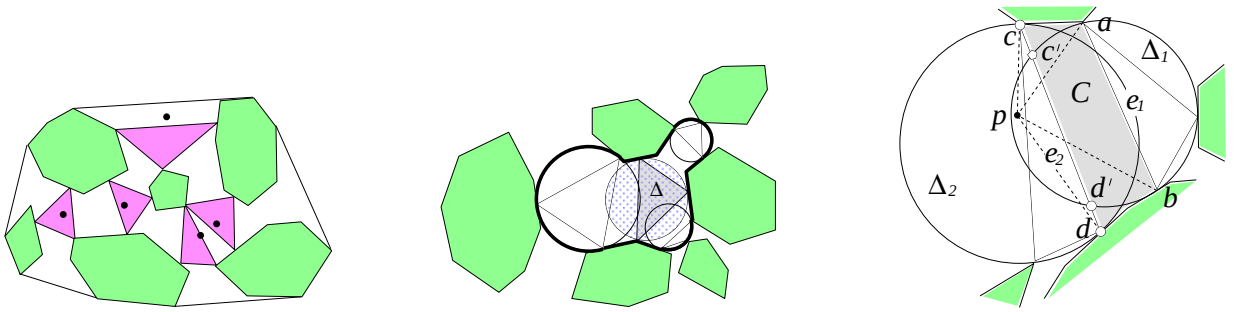


Figure 2. (a) The Delaunay triangulation of the compact Voronoi diagram. The free space between two adjacent junction triangles are corridors. (b) A junction triangle and its adjacent corridors and polygons. When Δ is locally Delaunay, $\tilde{\Delta}$ is covered by the adjacent corridors and circumcircles of its adjacent junction triangles, where the boundary of this region is thickened in the figure. (c) Proof of the local property.

global correctness of the diagram. When one of these conditions fails due to object motion, a local transformation (the 'edge-flip') is sufficient to restore local, and therefore global, correctness. The same principle was exploited to maintain the power diagrams of moving balls in [GZ98].

To pursue this idea in our current context, we first define a dual 'triangulation' of the (compact) Voronoi diagram. Recall that $\mathcal{F}(\mathcal{P})$ denotes the free space inside the convex hull of $\mathcal{P}$. A triangle is called a *junction triangle* if it is incident to three objects in $\mathcal{P}$. A *corridor* is a four-sided portion of the free space delimited by two polygons on two opposite sides. With slight abuse of notations, we call a cell decomposition with triangles and corridors as primitive cells a *triangulation*. A triangulation $\mathcal{T}(\mathcal{P})$ of $\mathcal{P}$ is a cell decomposition of $\mathcal{F}(\mathcal{P})$ into interior-disjoint junction triangles and corridors with all the vertices of $\mathcal{T}$ on polygon boundaries.

For a given polygon set $\mathcal{P}$, the number of junction triangles in a triangulation are specified by the following fact.

Fact 3.1. *For k disjoint convex polygons, if their convex hull contains h non-polygon edges, then any triangulation of their free space contains exactly $2k - h - 2$ junction triangles.*

We can form a triangulation of $\mathcal{F}(\mathcal{P})$ based on the compact Voronoi diagram of $\mathcal{P}$ as follows. For each junction vertex in the Voronoi diagram, its witness circle touches three polygons. We connect the three contact points to form a junction triangle corresponding to each junction vertex. It is easy to prove that the junction triangles thus formed do not intersect polygon interior and do not interpenetrate each other. If we remove these junction triangles from $\mathcal{F}(\mathcal{P})$, we will have a set of disconnected regions where each connected component is a corridor between two convex polygons. The corridors and junction triangles together form a triangulation that we will call the *Delaunay triangulation* of $\mathcal{P}$ and denote it by $\mathcal{D}(\mathcal{P})$ (Figure 2 (a)). If a junction triangle Δ is incident to three features f_1, f_2 and f_3, on different polygons, we say that the triplet (f_1, f_2, f_3) defines Δ. When objects start to move, the Delaunay triangulation changes combinatorially if such a triplet changes. In the following, we will show how to maintain $\mathcal{D}(\mathcal{P})$ when the objects in $\mathcal{P}$ move. We assume that during the motion, all the objects remain disjoint. As we will see later, we can use $\mathcal{D}(\mathcal{P})$ to detect collisions.

What is crucial for the maintenance of $\mathcal{D}(\mathcal{P})$ is its local certification property. For any junction triangle Δ, if Δ's circumcircle (denoted by $\tilde{\Delta}$) does not intersect the interior of any $P \in \mathcal{P}$, Δ is called a *Delaunay* triangle. Clearly, a triangulation is a Delaunay triangulation if and only if all of its junction triangles are Delaunay triangles. To state the local property we seek, we define a local Delaunay condition for a junction triangle. A corridor is said to be *adjacent* to a junction

triangle if they share an edge. Two junction triangles are *adjacent* if they are adjacent to the same corridor. Each junction triangle Δ then has up to three adjacent corridors and three adjacent triangles (Figure 2 (b)). The neighboring polygons for Δ are defined to be the polygons incident to it or to one of its adjacent junction triangles. We say that a junction triangle is *locally Delaunay* if its circumcircle does not intersect any of its neighboring polygons. Clearly, all the junction triangles of $\mathcal{D}(\mathcal{P})$ are locally Delaunay. We will show that the other direction also holds, which is the counterpart of the local certification property of the Delaunay triangulation for points.

Lemma 3.1 (Local Property). *If all the junction triangles of a triangulation $\mathcal{T}(\mathcal{P})$ are locally Delaunay, then $\mathcal{T}(\mathcal{P})$ is the Delaunay triangulation of $\mathcal{P}$.*

Proof: First, we observe that if a triangle Δ is locally Delaunay, then $\tilde{\Delta}$ is covered by the union of Δ, Δ's adjacent corridors, and the circumcircles of Δ's adjacent triangles (Figure 2 (b)).

Using the above fact, we will prove the local property by contradiction. Assume that $\mathcal{T}$ is not a Delaunay triangulation — then there must exist a point p interior to one of the polygons in $\mathcal{P}$ and a junction triangle $\Delta_1 \in T$ so that $p \in \tilde{\Delta}_1$. For an edge e_1 of Δ_1, denote by $\tilde{e}$ the sector of $\tilde{\Delta}_1$ bounded by the chord e_1. By the definition of triangulation, Δ_1 does not intersect any polygon interior. Therefore, $p \in \tilde{\Delta}_1 \setminus \Delta_1$, i.e., $p \in \tilde{ab}$ for a unique edge ab of Δ_1. Denote the angle apb by $\theta(p, \Delta_1)$. Among all the junction triangles whose circumcircles contain p, let Δ_1 be the one with the maximum $\theta(p, \Delta_1)$ and let e_1 be the edge of Δ_1 so that $p \in \tilde{e}_1$.

Now consider the corridor C adjacent to Δ_1 at e_1. Let the other edge bounding C to be e_2 and the junction triangle incident to e_2 be Δ_2. Since the corridor is the union of a set of triangles in $\mathcal{T}$, it cannot contain p. Thus, $p \in \tilde{e}_1 \setminus C$. This implies that $\tilde{e}_1$ intersects the edge e_2 since $\tilde{\Delta}_1$ does not intersect the polygons incident to Δ_1. By the local Delaunay property of Δ_1, $\tilde{\Delta}_1$ is covered by the union of corridors and the circumcircles of adjacent triangles, i.e., $\tilde{e}_1 \subset \tilde{\Delta}_2 \cup C$, or $\tilde{e}_1 \setminus C \subset \tilde{\Delta}_2$. Therefore, $p \in \tilde{\Delta}_2$. We can further conclude that p is not in $\tilde{e}_2$ because $\tilde{e}_1 \cap \tilde{e}_2 \subset C$. Suppose that the endpoints of e_2 are c and d and that e_2 intersects the boundary of $\tilde{\Delta}_1$ at points c' and d'. Then clearly the angle $\theta(p, \Delta_2) > \angle cpd > \angle c'pd' > \angle apb = \theta(p, \Delta_1)$, contradicting the maximality of $\theta(p, \Delta_1)$(Figure 2 (c)). □

By the above local property, to certify that a triangulation is the Delaunay triangulation, it is sufficient to certify that all the junction triangles are locally Delaunay, i.e., the circumcircle of each junction triangle does not intersect the interior of any of its neighboring polygons. Among the (up to six) neighboring polygons of a junction triangle Δ, three are incident to Δ and the others are incident to one of the adjacent junction triangles to Δ. We consider these two cases separately. In the following, we assume that Δ is defined by the triplet of features (f_1, f_2, f_3) where f_1, f_2, f_3 are features on P_1, P_2 and P_3, respectively.

To certify that $\tilde{\Delta}$ does not intersect the interior of its incident polygons it suffices, by convexity, to certify that $\tilde{\Delta}$ does not intersect any features adjacent to f_1, f_2 or f_3. All these certificates can be written as algebraic conditions in terms of the coordinates of (constant number of) polygon vertices. When a certificate fails, say, when $\tilde{\Delta}$ intersects the feature f_1', an adjacent feature of f_1, we then simply update the triplet of features from (f_1, f_2, f_3) to (f_1', f_2, f_3). The topological structure of $\mathcal{D}(\mathcal{P})$ is not affected by such events.

For those neighboring polygons that are not incident to Δ, suppose that Δ_1 is a junction triangle adjacent to Δ and it is incident to P_1, P_2 and P_4. To certify that $\tilde{\Delta}$ does not intersect P_4, it suffices to certify that $\tilde{\Delta}$ does not contain the point that is on Δ_1 on P_4 because $\tilde{\Delta}_1$ is disjoint from the interior of P_1, P_2 and P_4. When such a certificate fails, it must be the case that the circumcircles of Δ and Δ_1 are coincident. In other words, this happens when P_1, P_2, P_3, P_4 are cocircular. We call such events *cocircularity events.* We also note the special case when the junction triangle is on the

convex hull. In this case, the corresponding event is when the polygons become collinear and the common tangent line supports the convex hull of $\mathcal{P}$ or vice versa. For such *collinearity* events, we can watch each junction triangle with an edge on the boundary of $\mathcal{C}(\mathcal{P})$ to see when such a triangle degenerates and each pair of adjacent non-polygonal convex hull edges to see when such a triangle emerges. In the following, we will focus on the cocircularity events.

For Δ, Δ_1 to have the same circumcircle, they must share an edge, say e_1, which is incident to P_1 and P_2 — otherwise, we would contradict the fact that $\tilde{\Delta}$ and $\tilde{\Delta}_1$ do not intersect the interior of P_1 and P_2. After such a cocircularity event happens, the triangulation is no longer a valid Delaunay triangulation. To fix it, we simply delete the edge e_1 and add the other diagonal of the quadrangle formed by the union $\Delta \cup \Delta_1$ (Figure 3).

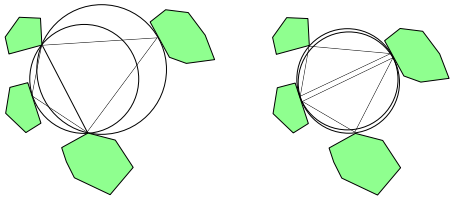


Figure 3. The cocircularity event and edge-flip operation.

We shall see that the above algorithm maintains a set of junction triangles that satisfy local Delaunay property. Further, note that only the collinearity events may change the number of junction triangles — we decrease or increase the number of junction triangles by one, depending on whether a convex hull edges appears or disappears. The other types of events do not affect the number of junction triangles. Therefore, we also maintain the right number of junction triangles. By Fact 3.1 and the local property, we can conclude that the above algorithm correctly maintains $\mathcal{D}(\mathcal{P})$, and thus the compact Voronoi diagram.

We have shown above how the compact Voronoi diagram can be maintained. Since the number of certificates is the sum of the number of junction triangles and the non-polygonal convex hull edges, the above structure has $O(k)$ certificates. The processing time for each event is $O(\log k)$, dominated by the processing cost of the event queue. As for the events processed, the above algorithm is output-sensitive in the sense that every event changes either the features that define a junction Voronoi vertex or the topology of the compact Voronoi diagram. Thus, we have,

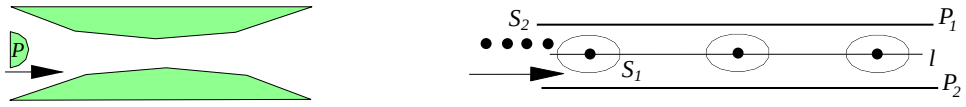
Theorem 3.2. *The compact Voronoi diagram of $\mathcal{P}$ can be maintained by a kinetic data structure in an output-sensitive manner. In the structure, the number of certificates is $O(k)$, and each event can be processed in $O(\log k)$ time.*

According to [BGH97], the structure we described is compact, responsive, and efficient. However, it is not local as one polygon may be involved up to $\Theta(k)$ certificates. Although the algorithm maintains the compact Voronoi diagram in an output-sensitive manner, we have not answered the question on how many events the data structure may need to process for algebraic polygon motions. To complete our analysis, in the next section, we will analyze the number of the combinatorial changes of the compact Voronoi diagram.

4 Combinatorial changes of the compact Voronoi diagram

In this section, we will study the number of combinatorial changes of the compact Voronoi diagram. For the analysis purpose, we assume that the polygons move rigidly in pseudo-algebraic motion with constant degree. That is, each certificate involving the same set of features can fail only constant number of times during the entire motion process. This assumption is very general as it includes the motions that can be represented in algebraic or rational functions with constant degree.

Observe that an event happens when there are four features cocircular or three collinear. This gives us an immediate upper bound of $O(n^4)$ for constant degree pseudo-algebraic motion. However, we will show a significantly smaller upper bound of $O(kn^2\beta(k)\beta(n))$. Here, $\lambda(n)$ is the maximum



(a) (b)

Figure 4. Lower bound constructions: (a) shows an example with $\Omega(n^2)$ changes to the compact Voronoi diagram. (b) shows an example with $\Omega(k^2)$ changes to a pair of objects.

length of a (n, s) Davenport-Schinzel sequence for some constant s, and $\beta(n) = \lambda(n)/n$ is an extremely slowly growing function and can be regarded as close to a constant for all reasonable values of n. On the other hand, there are examples to show an $\Omega(n^2)$ lower bound by three convex objects (Figure 4 (a)).

Now, our main task is to prove the following theorem.

Theorem 4.1. *Suppose that $\mathcal{P}$ is a set of k disjoint convex polygons with n vertices in total. When all the polygons in $\mathcal{P}$ move algebraically and without colliding with each other, the compact Voronoi diagram changes $O(n\lambda(n)\lambda(k)) = O(kn^2\beta(n)\beta(k))$ times.*

In [HKK92], a similar upper bound is proved for the number of changes of the Voronoi diagram of k sets of points, each moving independently and rigidly. We will use similar techniques to prove our upper bound. However, the presence of edges adds complexity, and additional insights are required to complete the proof.

We first give an example to show the difference from the points case. In the proof of an $O(k^3\beta(k))$ bound on the number of changes of the Voronoi diagram of k moving points, the key step is to bound the number of changes involving two points by $O(k\beta(k))$. This fact, however, is no longer true for polygons. Consider the example shown in Figure 4 (b). When the $k/2$ points in S_2 move from $x = -\infty$ to $x = +\infty$, there will be $\Theta(k^2)$ changes of the Voronoi edges incident to P_1 and P_2. However, as we shall see later, still there is a way to charge each change to a carefully chosen polygon pair so that no pair receives too many charges.

To proceed, we first consider how Voronoi vertices move when polygons move. Consider three convex polygons, i.e., $\mathcal{P} = \{P_1, P_2, P_3\}$. We write that $n_i = |P_i|$, where $1 \le i \le 3$, and $n = n_1 + n_2 + n_3$. For three objects, a combinatorial change can happen to $\mathcal{D}(\mathcal{P})$ only when the triplets of features that define the Voronoi vertices change. For the number of combinatorial changes for constant degree algebraic motion, we have the following bound.

Lemma 4.2. *The compact Voronoi diagram of three convex polygons changes $O((n_1n_2 + n_1n_3 + n_2n_3)\beta(n))$ times.*

Proof: (Sketch) Consider the time when four features are co-circular. There must be two features from the same polygon, say P_1. Further, by the convexity, these two features must be an edge, say e, and an endpoint, say p, of e. By considering the family of circles tangent to e at p, we can bound the number of the events that involve p and e by $O(\lambda(n_2) + \lambda(n_3))$. The bound is summed over all such feature pairs. □

Because of the way that we parameterize the bisector, we also need the following fact.

Lemma 4.3. *The distance function $\delta(P_1, P_2)$ consists of $O(n_1n_2)$ rational arcs with constant degree when P_1,P_2 moves algebraically with constant degree.*

For three polygons P_1, P_2 and P_3, recall that $S_{12,3}$ is the set of points that are on π_{12} and shaded by P_3. The parameters of this shaded set, $\tilde{S}_{12,3}$, may be of the forms $\emptyset$, $(\infty, a]$, $[b, +\infty)$,

$(-\infty, a] \cup [b, +\infty)$, $[a, b]$, or $(-\infty, +\infty)$. We define the function $\phi_{12,3}(t)$ as follows. If at time t, π_{12} is half-shaded by P_3 at a, then $\phi_{12,3}(t)$ is defined to be a. Otherwise, it is undefined. Since an endpoint of $\tilde{S}_{12,3}$ corresponds to the parameter value of a Voronoi vertex when considering P_1, P_2 and P_3 only, Lemma 4.2 and 4.3 say that $\phi_{12,3}$ consists of $O((n_1n_2 + n_1n_3 + n_2n_3)\beta(n))$ pieces of rational arcs. Likely, we define the function $\phi_{ij,l}$ for each triplet i, j, l.

For a pair of polygons P_i and P_j, we have a family of functions $\Phi_{ij} = \{\phi_{ij,l} \,|\, l \neq i, j\}$. Let $\Gamma(\Phi)$ denote the upper envelope of a set of functions Φ. We first show that

Lemma 4.4. *Each cocircular event can be charged to a break point on $\Gamma(\Phi_{ij})$ or the overlay between $\Gamma(\Phi_{ij})$ and $-\Gamma(\Phi_{ji})$, for some $i \neq j$.*

For the moment, let us assume that the above lemma is true and prove Theorem 4.1.

Proof: (Theorem 4.1). As we have already discussed, there are two types of events: type I, when the feature triplets defining Voronoi vertices change, and type II, when four polygons become cocircular. By Lemma 4.2, the number of type I of events is bounded by:

$$\textstyle\sum_{i,j,l}(n_in_j + n_in_l + n_jn_l)\beta(n) = O(kn^2\beta(n))\,.$$

For type II events, by Lemma 4.4, they can be charged to break points on the lower or upper envelopes of Φ_{ij}'s or their overlay. Since each $\phi_{ij,l}$ consists of $O((n_in_j + n_in_l + n_jn_l)\beta(n))$ pieces of rational arcs. The complexity of $\Gamma(\Phi_{ij})$ is then bounded by:

$$\beta(k)\textstyle\sum_l(n_in_j + (n_i + n_j)n_l)\beta(n) = O((kn_in_j + (n_i + n_j)n)\beta(n)\beta(k))\,.$$

The overlay between two envelopes has the same order of complexity. Thus, the number of cocircular events is bounded by $\sum_{i,j}(O((kn_in_j + (n_i + n_j)n)\beta(n)\beta(k))) = O(kn^2\beta(n)\beta(k))$. □

Now, the only piece left is the proof of Lemma 4.4.

Proof: (Lemma 4.4). (Sketch) Suppose that at time t, a cocircularity event happens to P_1, P_2, P_3 and P_4. We claim that among those four polygons, there always exist two, say P_1 and P_2, so that $\tilde{S}_{12,3}$ and $\tilde{S}_{12,4}$ are not closed intervals with the form $[a, b]$. This claim can be argued by Fact 2.1, and we omit the detail in this abstract.

Suppose v is the coincident Voronoi vertex at time t. Let $x = \zeta_{12}(v)$. By Fact 2.2, at time t, there cannot be any other P_i ($i \neq 1, 2, 3, 4$) so that x in the interior of $\tilde{S}_{12,i}$. Since $\tilde{S}_{12,3}$ are not closed intervals, either $\phi_{12,3}(t) = x$ or $\phi_{21,3}(t) = -x$. The same argument applies to $\phi_{12,4}(t)$ and $\phi_{21,4}(t)$. The fact that v is not shaded by any other P_i allows us to charge such an event either to a break point on $\Gamma(\Phi_{12})$ or $\Gamma(\Phi_{21})$ or to an intersection between $\Gamma(\Phi_{12})$ and $-\Gamma(\Phi_{21})$. □

5 Applications

In this section, we briefly discuss some applications of the above data structure. In the above presentation, one important issue left unspecified in our method is the way to handle the corridors. We will present different structures dependent on the application requirements.

5.1 Collision detection

A major motivation to maintaining a decomposition of free space is to detect collision for moving objects [EGSZ99, BEG+99]. If for each corridor, we add an inner bi-tangent line between the two convex chains bounding the corridor, we can detect collision between the objects involved (refer to [EGSZ99] for why we prefer tangent based separation to the separation based on the closet pair). In [EGSZ99], efficient hierarchical methods are developed to reduce the number of events associated with tracking tangents. Those methods can be used in our setting as well.

5.2 Retraction motion planning

The compact Voronoi diagram can be used to do retraction motion planning (so that the robot finds a path that stays maximally far from the obstacles). In retraction motion planning, we need to know the narrowest passage between two convex polygons. For this purpose, we may maintain the closest pair of features between two convex chains in a corridor. In [LC91, Mir97], there are local conditions given to check if a pair of features is the closest pair. It is not hard to see that such a condition can be used to certify and then maintain the nearest pair.

6 Conclusion

We have shown how to maintain a partition of the free space outside k moving convex polygons in the plane into triangles and corridors. In each cell of this partition the closest obstacle is one of the two or three polygons defining the corridor or triangle respectively. Our structure continuously maintains $O(k)$ polygon pairs among which must be the closest pair of polygons. With the addition of a simple corridor collision test, as outlined above, the kinetic compact Voronoi diagram subsumes both the broad and narrow phases as commonly defined in the collision detection literature.

Unlike more classical methods, our structure can easily accommodate deforming obstacles, as long as they stay convex. Though the details remain to be worked out, an extension of our structure to 3D also looks feasible.

References

[BEG+99] J. Basch, J. Erickson, L. J. Guibas, J. Hershberger, and L. Zhang. Kinetic collision detection for two simple polygons. In *Proc. 9th ACM-SIAM Sympos. Discrete Algorithms*, pages 102–111, 1999.

[BGH97] J. Basch, L. J. Guibas, and J. Hershberger. Data structures for mobile data. In *Proc. 8th ACM-SIAM Sympos. Discrete Algorithms*, pages 747–756, 1997.

[EGSZ99] Jeff Erickson, L. J. Guibas, Jorge Stofi, and L. Zhang. Separation-sensitive kinetic collision detection for convex objects. In *Proc. 9th ACM-SIAM Sympos. Discrete Algorithms*, 1999.

[GMR92] L. J. Guibas, J. S. B. Mitchell, and T. Roos. Voronoi diagrams of moving points in the plane. In G. Schmidt and R. Berghammer, editors, *Proc. 17th Internat. Workshop Graph-Theoret. Concepts Comput. Sci.*, volume 570 of *Lecture Notes Comput. Sci.*, pages 113–125. Springer-Verlag, 1992.

[GZ98] L. Guibas and L. Zhang. Euclidean proximity and power diagram. In *Proc. 10th Canadian Conference on Computatoinal Geometry*, 1998.

[HKK92] 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. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 110–119, 1992.

[Lat91] J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, 1991.

[LC91] M. C. Lin and J. F. Canny. Efficient algorithms for incremental distance computation. In *Proc. IEEE Internat. Conf. Robot. Autom.*, volume 2, pages 1008–1014, 1991.

[Mir97] B. Mirtich. V-Clip: fast and robust polyhedral collision detection. Technical Report TR-97-05, Mitsubishi Electrical Research Laboratory, 1997.

[MKS96] M. McAllister, D. Kirkpatrick, and J. Snoeyink. A compact piecewise-linear Voronoi diagram for convex sites in the plane. *Discrete Comput. Geom.*, 15:73–105, 1996.

[PS90] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, 3rd edition, October 1990.