CAR-TR-699 CCR 93-10705
CS-TR-3201 January 1994

On the Area of Overlap of Translated Polygons

David M. Mount
Department of Computer Science and
Institute for Advanced Computer Studies
University of Maryland

Ruth Silverman
Department of Computer Science
University of the District of Columbia
and
Center for Automation Research
University of Maryland

Angela Y. Wu
Department of Computer Science and Information Systems
The American University

Abstract

Given two simple polygons P and Q in the plane and a translation vector $t \in R^2$, the *area-of-overlap* function of P and Q is the function $\mathrm{Ar}(t) = \mathrm{Area}(P \cap (t + Q))$, where $t + Q$ denotes Q translated by t. This function has a number of applications in areas such as motion planning and object recognition. We present a number of mathematical results regarding this function. We also provide efficient algorithms for computing a representation of this function, and for tracing contour curves of constant area of overlap.

The support of the National Science Foundation under grant CCR 93-10705 is gratefully acknowledged, as is the help of Sandy German in preparing this paper. A preliminary version of this paper appeared in *Vision Geometry II*, R.A. Melter and A.Y. Wu, Editors, Proc. SPIE 2060, 1993, 254–264.

1 Introduction

An important geometric problem involving planar shapes is whether two simple polygons intersect one another. If the polygons do intersect it is often useful to acquire more information regarding the nature of the degree of overlap. One measure of the degree of overlap is the area of overlap between the two polygons. In many applications the placement of one or both of the polygons is subject to translation. In this case it is natural to consider how the area of overlap of the two polygons varies as a function of their mutual relationship. In this paper we analyze a number of mathematical and computational properties of the area of overlap of polygons under translation.

Let P be a simple polygon, that is, a closed connected set of points in the plane bounded by a closed polygonal Jordan curve. Let $\mathrm{Int}(P)$ denote the interior of P and $\mathrm{Ver}(P)$ denote its vertex set, and $\mathrm{Bnd}(P)$ denote its boundary. These are to be thought of as geometric sets in the plane, that is, as sets of points in R^2. (Most of the results of this paper are applicable to the more general case of closed bounded sets with polygonal boundaries, that are not necessarily simply connected.) Given a polygon P in the plane, and vector $t \in R^2$, the *translate* of P by t, denoted $t + P$, is the set $\{t + p \mid p \in P\}$. Let $-P = \{-p \mid p \in P\}$, and define $t - P$ to be $t + (-P)$. Given two simple polygons P and Q, the *area-of-overlap function* is the function $\mathrm{Ar} : R^2 \to R$ given by $\mathrm{Ar}(t) = \mathrm{Area}(P \cap (t+Q))$. It suffices to assume that only one of the two polygons has been translated because $\mathrm{Area}((s + P) \cap (t + Q)) = \mathrm{Area}(P \cap ((t - s) + Q))$. For arbitrary sets P and Q in the plane, we define their *sum* (also called *vector sum* or *Minkowski sum*) $P + Q = \{p + q \mid p \in P, q \in Q\}$.

There are a number of interesting applications and interpretations of this function. In motion planning in which two objects can be moved by a sequence of translations, the set of *feasible placements* of the two objects, defined to be the set of nonoverlapping placements, is just the set over which the area-of-overlap function is equal to zero. Combinatorial properties of the region of zero overlap for two simple polygons were studied by Pollack, Sharir and Sifrony [12]. Kedem, Livne, Pach and Sharir [9] considered the situation of a single moving convex polygon amidst a set of stationary nonintersecting convex polygons. Other applications of zero overlap placements include packing and covering by translates of polygons [11, 10]. The general area-of-overlap function is a natural generalization of the area of zero overlap when objects are defined to be "fuzzy" and the goal is to minimize the area of overlap. The area of overlap can be also be viewed as a convolution between two functions which are unit valued over the interiors of P and Q.

In this paper we consider a number of mathematical and computational results involving the area-of-overlap function. We show that the area-of-overlap function of two simple polygons is a continuous, piecewise polynomial surface of degree at most two. We present bounds on the combinatorial worst case complexity of this surface given the number of sides of the two polygons. We also present an efficient algorithm for computing a representation of the area-of-overlap function. Finally we present an algorithm for efficiently tracing a contour curve of constant area of overlap.

2 Object Recognition by Probing

The application which brought us to the area-of-overlap problem was a method of performing model-based object recognition in computer vision by means of a method called *probing*. The problem of object recognition by probing has been well studied recently by computational geometers. The early work was in the area of *finger probes* where an object is identified by shooting a ray from infinity until it contacts the object's boundary [4, 5]. The probing paradigm was generalized by Skiena and others to include other classes of probes [6, 14, 15, 16].

We consider another version of the probing paradigm that seems to be relevant to computer vision. We assume that a single polygonal object P, called the *target*, has been translated to an unknown position within a bounding rectangular region, called the *image*. We assume that there is a local operator that is capable of determining whether some point (x, y) of the image lies within the interior of the translated polygon. In this case the point (x, y) is called a *hit* and otherwise it is a *miss*. Each such point (x, y) is called a *probe*. The problem is how to use the fewest probes most accurately to fix the location of the target. We assume that we have at least one initial hit to start the process off. The use of probing for object recognition has been studied in [1, 2].

Each possible placement of the polygon is specified by a translation vector t. Given any set of hits and misses, the locus of placements of P that are consistent with this set forms a *polygonal domain* in the plane, that is, a possibly disconnected region of the plane whose boundary consists of a finite number of line segments. To see this, if a point v is a hit, then (assuming that there is only one copy of P in the image) we can infer that $v \in t + P$, implying that $t \in v - P$. Similarly, if the point v is a miss, then we can infer that $t \notin v - P$. Given a set of probes, $v_1, v_2, \ldots, v_h$ being hits and $u_1, u_2, \ldots, u_m$ being misses, we infer that the set of feasible values of t (possible placements of P) is given by the set

$$\left(\bigcap_{1 \leq i \leq h} (v_i - P) \right) \Bigg\backslash \left(\bigcup_{1 \leq i \leq m} (u_i - P) \right),$$

(where $A \setminus B$ denotes the set theoretic difference of A and B). This set is called the *feasible region*. Since it is formed from set operations on simple polygons it is a polygonal domain. Thus, for a given number of probes, we want to minimize some measure of the size of the feasible region, such as its area or diameter.

Given such a region of feasible placements, a natural question to ask is where to place the next probe so that it provides the greatest amount of information. One possible criterion is to select the next probe to minimize the *area* of the feasible region. Since we do not know the result of the next probe, the probe should be chosen so that, in the worst case, it reduces the area of the feasible region as much as possible. If F denotes the current feasible region, and v is the choice for the next probe, then the next feasible region will be $F \cap (v - P)$ if the probe hits and $F \setminus (v - P)$ if the probe misses. In the worst case, the area of the feasible region after this probe is

$$\max(\mathrm{Area}(F \cap (v - P)), \mathrm{Area}(F \setminus (v - P))).$$

Since increasing the area of one term of the maximum decreases the area of the other, we minimize the worst case area by selecting v so that $\mathrm{Area}(F \cap (v - P))$ is as large as possible but not greater than $\mathrm{Area}(F)/2$. In other words, we seek a placement of $-P$ such that its overlap with F is equal to $\mathrm{Area}(F)/2$, and if no such placement exists, then one that maximizes the area of overlap with $-P$ and F.

In general there may be many placements that satisfy these conditions, and so other criteria (such as the combinatorial complexity of the resulting feasible region) may need to be considered. For this reason, computing the entire area-of-overlap function may be preferred to the simpler task of computing an arbitrary single-placement that satisfies these conditions.

3 Combinatorial and Analytic Structure

In this section we consider the structure of the area-of-overlap function, both its analytic structure and combinatorial structure. It is easy to see that the area-of-overlap function is nonzero over a bounded region of the plane. We begin with a series of easy observations.

CLAIM 3.1 *The subset of R^2 for which* $\mathrm{Ar}(t)$ *is nonzero is the Minkowski sum* $\mathrm{Int}(P) - \mathrm{Int}(Q)$.

PROOF: P and $t+Q$ have a nonzero overlap if and only if there are interior points $p \in \mathrm{Int}(P)$ and $q \in \mathrm{Int}(q)$ such that $p = t + q$, or equivalently $t = p - q$. This is true if and only if $t \in \mathrm{Int}(P) - \mathrm{Int}(Q)$. □

CLAIM 3.2 *If P and Q are simple polygons then the subset of R^2 over which the function* $\mathrm{Ar}(t)$ *is nonzero is path connected. The subset of R^2 over which the function is zero need not be connected.*

PROOF: Let t and t' be points in R^2 for which the area-of-overlap function is nonzero. Observe that if P and $t + Q$ have a nonzero overlap then there are points $p \in \mathrm{Int}(P)$ and $q \in \mathrm{Int}(Q)$ such that $p = t + q$. Similarly, if P and $t' + Q$ have a nonzero overlap then there are points $p' \in P$ and $q' \in Q$ such that $p' = t' + q'$. Because the interior of Q is connected there is a path connecting q' to q through the interior of Q. We can parameterize the points along this path (e.g. by path length) yielding continuous functions $x(s)$ and $y(s)$ for $s \in [0, 1]$ such that $q = (x(0), y(0))$ and $q' = (x(1), y(1))$ and such that $(x(s), y(s)) \in \mathrm{Int}(Q)$ for all $s \in [0, 1]$. Let $t(s) = t + q - (x(s), y(s))$. The connected set of translates $t(s) + Q$, for $0 \le s \le 1$, contains the point $t + q - (x(s), y(s)) + (x(s), y(s)) = t + q = p$, and so overlaps P. In the final translate, $t(1) + Q$, the point p coincides with the point q'. Similarly, because

Int(P) is connected, we can perform an analogous path translation of P to bring the points p' and q' into coincidence while maintaining nonzero overlap.

To prove that the subset over which the function is zero need not be connected, let Q be a unit square. Consider a square of side length 3 containing a hole slightly larger than a unit square. Although this is not a simple polygon, it can be made one by connecting the hole to the outer boundary by a narrow channel. Call the resulting polygon P. The area of overlap is zero when Q is placed inside the hole, and is zero when Q is placed well outside of P, but Q cannot be moved from one placement to the other without incurring a nonzero overlap. $\square$

CLAIM 3.3 *The area-of-overlap function is continuous.*

PROOF: This is obvious since the overlap itself is formed from a finite set of simple polygons, and any infinitesimal translation can affect the dimensions of each such polygon only infinitesimally. $\square$

Define a *placement* to be a pair $(P, t + Q)$. It is easy to see that for many placements a small perturbation in t induces a smooth variation in the area-of-overlap function, however there are certain critical placements at which the nature of the overlap can change suddenly. For example, if t lies on the boundary of $P - Q$, where the area function changes from nonzero to zero, the function may not behave smoothly. We consider under what circumstances such changes occur.

A placement $(P, t+Q)$ is said to be *critical* if either a vertex of P intersects the boundary of $t + Q$ or a vertex of $t + Q$ intersects the boundary of P. Let Crit(P, Q) denote the set of such placements. Define the *combinatorial type* of a placement to be the set of pairs of edges/vertices of P and edges/vertices of Q that intersect one another. We say that two placements given by s and t are *equivalent* if there is a path π from s to t such that the combinatorial type is constant along the path. The equivalence classes of this relationship subdivide the plane into regions. Noncritical regions of this subdivision have dimension 2. Since a critical region involves the intersection of at least one vertex of one polygon with an edge of the other, the dimension of a critical region cannot be greater than 1. Recall that $\mathrm{Ver}(P)$ denotes the set of vertices of P and $\mathrm{Bnd}(P)$ denotes the boundary of P.

LEMMA 3.1 *The locus of critical placements,* $\mathrm{Crit}(P, Q)$*, is given by the union*

$$(\mathrm{Ver}(P) - \mathrm{Bnd}(Q)) \cup (\mathrm{Bnd}(P) - \mathrm{Ver}(Q)).$$

PROOF: A placement given by t is critical if and only if it corresponds to an intersection between a vertex of P and the boundary of Q, or vice versa. In the former case this is equivalent to saying that there is $p \in \mathrm{Ver}(P)$ and $q \in \mathrm{Bnd}(Q)$ such that $p = t + q$, or equivalently $t = p - q$, meaning that $t \in \mathrm{Ver}(P) - \mathrm{Bnd}(Q)$. The other case is analogous. □

LEMMA 3.2 *Let* $t = (t_x, t_y)$*. Let* $E(t)$ *denote the set of all equivalent placements. For all points in* $E(t)$*, the area-of-overlap function is given by a polynomial in* t_x *and* t_y *of maximum degree 2.*

PROOF: Let us assume that P and Q share no parallel edges. In general the theorem holds by applying a continuity argument to the limiting case in which edges approach being parallel. Consider the placement $(P, t + Q)$. It suffices to show that, for all sufficiently small translations Δ such that the placements $(P, t + Q)$ and $(P, t + \Delta + Q)$ are combinatorially equivalent, the difference of area $\mathrm{Ar}(t + \Delta) - \mathrm{Ar}(t)$ is a polynomial of degree 2 as a function of Δ's coordinates, Δ_x and Δ_y.

If a vertex of P overlaps a vertex of $t + Q$ then it follows that $E(t)$ consists of just the single point t, and the lemma is trivially true. Let us assume this is not the case. Consider the finite point set S consisting of the union of the vertex set of P, the vertex set of $t + Q$ and the set of all points defined by the intersection of the interiors of two edges of P and $t + Q$. It is easy to see that under our hypotheses this is a finite set of isolated points. Since the set is finite, we can subdivide the plane into a finite number of polygonal cells such that (1) each cell either contains a single edge of one of the polygons, or contains a single point of S and the edges of the polygon(s) incident to this point, and (2) the vertices of the cells

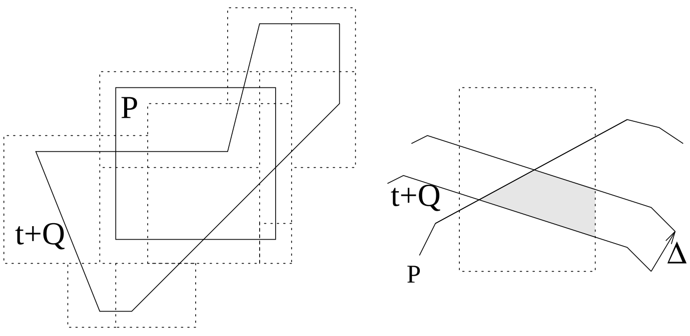

Figure 1: Decomposition of the area of overlap.

do not intersect the boundary of either polygon. An example of such a subdivision is shown in Fig. 1(a).

Since the cells have a particularly simple structure we can calculate their areas individually and sum them. Since vertices of the cells do not intersect the boundaries of the polygons, and since the vertices and intersection points of the polygon boundaries do not intersect the boundaries of the cells, any infinitesimal perturbation of the translation vector Δ does not affect the incidence relations between the polygons and the subdivision. By restricting $t+\Delta$ to equivalent placement we do not affect the incidence relations between the two polygons.

A cell that contains no polygon boundary, or that contains only the boundary of the unmoving polygon P, does not change its contribution to the area of overlap as Δ varies. For a cell that contains only one edge of the boundary of $t+Q$ or for a cell that contains an intersection point of the boundaries of P and $t+Q$, it is easy to see that the change in area can be expressed as a finite sum of trapezoids (degenerating possibly to triangles), where the angles of the sides remain fixed, but the distance between the bases increases linearly with Δ_x and Δ_y. See Fig. 1(b). It follows that the lengths of the bases of the trapezoids also vary linearly with Δ_x and Δ_y, so the area of the resulting trapezoid varies quadratically in Δ_x and Δ_y. Summing over all such cells completes the proof. $\square$

For a fixed area value α, the set of *contour lines* arising by considering all placements t such that $\mathrm{Ar}(t) = \alpha$ is a set of planar curves formed from pieces that are polynomials of degree at most 2. Thus we have the following.

COROLLARY 3.1 *Given simple polygons P and Q, for a fixed area α, the locus of placements t such that the area of overlap is equal to α, $\{t \in R^2 \mid \mathrm{Ar}(t) = \alpha\}$, consists of curves that are (possibly degenerate) piecewise conics.*

THEOREM 3.1 *Let P and Q be two simple polygons with m and n sides respectively. The area-of-overlap function $\mathrm{Ar}(t)$ is a piecewise polynomial function of degree at most 2 over R^2. The number of pieces is $O((mn)^2)$. This bound is tight in the worst case. If both P and Q are convex then the number of pieces is $O(m^2+n^2+mn\min(m,n))$, and if further, P and Q are equal up to translation then the number of pieces is $O(m^2)$.*

PROOF: Lemma 3.2 establishes that the area-of-overlap function $\mathrm{Ar}(t)$ is a piecewise polynomial function of degree at most 2 in the coordinates of t. Lemma 3.1 establishes that the regions of the equivalence classes are bounded by a Minkowski sum of the vertices of P and the boundary of $-Q$, and vice versa. A Minkowski sum of the m vertices of P and the n edges of $-Q$ is a line segment arrangement with mn segments. Such an arrangement can have at most $O((mn)^2)$ faces.

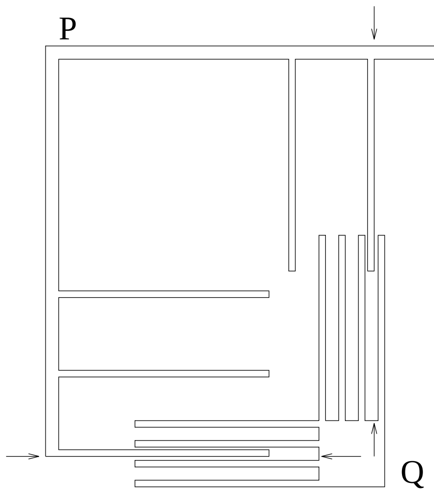

Figure 2: Worst case example.

To see that the bound is tight consider the example shown in Fig. 2, which is due to Pollack, Sharir, and Sifrony [12]. The polygon P consists of two "combs" each with m "teeth", such that the teeth of the two combs are perpendicular to each other. Q consists of two combs each with $n+1$ teeth, and hence n "gaps" between the teeth. Clearly P and

Q consist of $O(m)$ and $O(n)$ sides respectively. It is evident from the figure that for each of the m^2 possible choices of one horizontal tooth and one vertical tooth from P and for each of the n^2 possible choices of one horizontal and one vertical gap from Q, there is a placement of P and Q such that the designated teeth of P protrude within the designated gaps of Q, implying zero overlap. Any two such placements are combinatorially inequivalent, because any motion between the two placements must pass through a placement of nonzero overlap. Since all choices are independent, there are $O((mn)^2)$ connected components of zero overlap, and hence the number of pieces is at least this large.

If both P and Q are convex, then the arrangement defining the set of critical placements has a particular structure. In particular, the Minkowski sum $\mathrm{Ver}(P) - \mathrm{Bnd}(Q)$ consists of m copies of identical n-sided convex polygons. Since the boundaries of two translates of a convex polygon can intersect at most twice, there can be at most $O(m^2)$ intersection points. Similarly, there can be at most $O(n^2)$ intersection points in the arrangement of $\mathrm{Bnd}(P) - \mathrm{Ver}(Q)$. Finally, in the union of the two arrangements, for any pair of the mn convex polygons, one from each arrangement, there can be at most $\min(m, n)$ intersections (since each edge of one polygon can intersect at most two edges of the other polygon). Thus the total number of intersections in the arrangement is $O(m^2 + n^2 + mn \min(m, n))$. The total number of vertices in the arrangement is dominated by this quantity. Finally, by Euler's formula, the number of faces in the arrangement is a linear function of the number of vertices in the arrangement.

Finally, if P and Q are translates of one another, then the above analysis holds, except that any pair of polygons, one chosen from each of the arrangements, can intersect in at most two points. Thus the $\min(m, n)$ term is replaced by 2. Since $m = n$, this gives $O(m^2)$. $\square$

4 Area Computation

In this section we consider the problem of computing a representation of the area-of-overlap function. Recall that $\mathrm{Crit}(P, Q)$ denotes the line segment arrangement

$$(\mathrm{Ver}(P) - \mathrm{Bnd}(Q)) \cup (\mathrm{Bnd}(P) - \mathrm{Ver}(Q)),$$

and let $\mathrm{Arr}(P, Q)$ denote the combinatorial complexity (number of edges) in this arrangement. The line segment arrangement $\mathrm{Crit}(P, Q)$ is formed from m copies of the n segments of Q, plus n copies of the m segments of P, implying that there are $O(mn)$ segments in total in the arrangement, and hence its combinatorial complexity, $\mathrm{Arr}(P, Q)$, is at most $O((mn)^2)$. By Lemmas 3.1 and 3.2 each face of this arrangement can be associated with the coefficients of a polynomial in x and y of degree at most 2. This polynomial defines the area-of-overlap function for each point within the face. This provides a representation of the area-of-overlap function whose size is $O(\mathrm{Arr}(P, Q))$.

Combining the proofs of Lemma 3.2 and Theorem 3.1, we can derive a naive algorithm with running time

$$O(mn \log(mn) + \mathrm{Arr}(P, Q)((m + n) \log(m + n) + mn)) \leq O((mn)^3)$$

for computing the area-of-overlap function. Using standard algorithms from computational geometry, we can compute the line segment arrangement $\mathrm{Crit}(P, Q)$ in $O(mn \log(mn) + \mathrm{Arr}(P, Q))$ time [3]. By Euler's formula, the number of faces in this arrangement is proportional to its complexity. The dual graph whose vertices are the faces of the arrangement can be computed in the same time bound needed to compute the arrangement.

For each of the $O(\mathrm{Arr}(P, Q))$ faces of the arrangement, we construct a representative placement of P and Q for this equivalence class, by taking any point t within the interior of this face and constructing the line segment arrangement defined by the boundaries of P and $t + Q$. This *single-placement arrangement* (not to be confused with the arrangement of critical placements, $\mathrm{Crit}(P, Q)$) contains $m + n$ segments and can be constructed in $O((m + n) \log(m + n) + mn)$ time [3]. We can compute a subdivision satisfying the conditions of Lemma 3.2 within this same time bound, by first computing trapezoidal decomposition of the arrangement in the same time bound [3], then "fattening" each edge of the arrangement by a sufficiently small amount (forming a set of corridors that surround the boundaries of the polygons), and finally subdividing each of these corridors into polygonal regions that separate nonincident elements. Because each of the resulting regions contains at most a constant number of edges and at most one vertex of the arrangement, we can compute the contribution of each region of the subdivision to the area of overlap in $O(1)$ time. The time for this computation is dominated by the time to compute the single-placement arrangement.

The total complexity of this method is $O(mn\log(mn) + \mathrm{Arr}(P,Q))$ to compute and traverse the arrangement of critical placements, and then for each of the $O(\mathrm{Arr}(P,Q))$ faces of $\mathrm{Crit}(P,Q)$ we spend $O((m+n)\log(m+n)+mn)$ time to compute the area function. Thus the total time is $O(mn\log(mn) + \mathrm{Arr}(P,Q)((m+n)\log(m+n)+mn))$.

The main result of this section is to show that we can shave off the factor of $O((m+n)\log(m+n)+mn)$ in the running time of this procedure. This is done by incrementally updating the contribution to the area-of-overlap function in $O(1)$ time per face of the arrangement of critical placements (rather than computing it from scratch each time).

THEOREM 4.1 *Given two polygons P and Q, of m and n sides respectively,*

(i) a representation of the area-of-overlap function for these two polygons can be computed in $O(mn\log(mn)+\mathrm{Arr}(P,Q)) \leq O((mn)^2)$ time, where $\mathrm{Arr}(P,Q)$ is the complexity of the arrangement of critical placements of P and Q,

(ii) the representation of the function has space complexity $O(\mathrm{Arr}(P,Q)) \leq O((mn)^2)$, and

(iii) given this representation, the area of overlap of P and $t+Q$ can be computed in $O(\log(n+m))$ time, given any translation vector t.

PROOF: As before we construct the arrangement of critical placements, $\mathrm{Crit}(P,Q)$, and as we are doing so, we label each edge with a pair of indices, indicating the vertex of P and edge of Q, or vertex of Q and edge of P that gave rise to this edge. This can be done within the same asymptotic time bound as before.

As before we traverse the faces of this arrangement, and for each face we compute the coefficients of the area-of-overlap function. We start with the external face of the arrangement, since the area-of-overlap function is just the zero function. As we walk around the line segment arrangement (e.g. by a depth-first traversal of its dual graph), suppose that we walk from a face f to a neighboring face f' across an edge that is labeled with vertex v of P and edge e of Q. The transition in the area-of-overlap function as the translation vector crosses from f to f' arises because the edge e has been translated so that it crosses over vertex v. The edge separating f from f' in the arrangement corresponds to a set of translations that cause edge e to intersect vertex v. In Fig. 3(a) Crit(P, Q) is shown, and the corresponding translation is shown in (b).

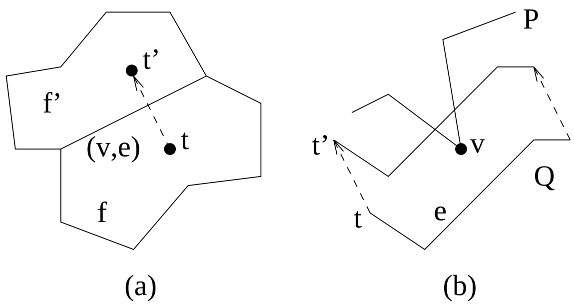

Figure 3: Critical transition.

In order to determine the incremental change in the area-of-overlap function as we move from f to f', we first observe that it suffices to determine the incremental change in the function in any small neighborhood about any transition point, since the function is a constant within each face of the arrangement of critical placements. To do this, let us consider a pair of placements t in f, and t' in f', which we assume to be placed arbitrarily close to each other (see below). Let R denote a polygon that contains only a segment of e, v, and segments of the edges incident to v. Also assume that R is chosen so that, excluding the interior of R, all translations between t and t' are combinatorially equivalent (including the intersections between P and Q with the boundary of R). See Fig. 4(a). It is clear that if t and t' are close enough to one another, and the line segment between t and t' does not pass through any critical vertex, then such a polygon R can be found in constant time by a local analysis the polygons P, $t + Q$ and $t' + Q$ around the critical point.

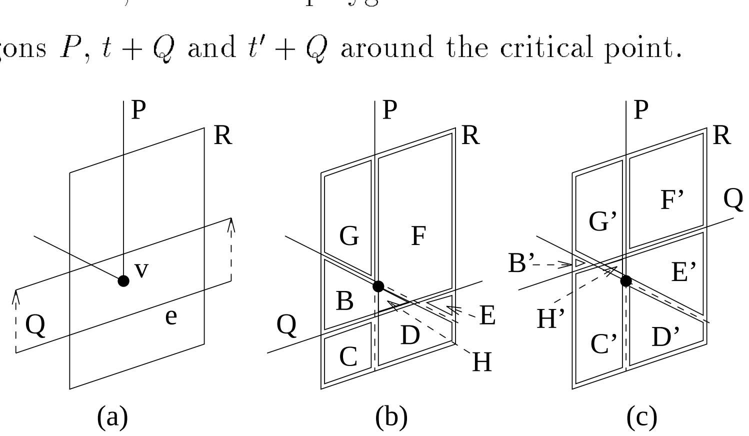

Figure 4: Updating the area function across a critical placement.

Within the interior of R there are three edges, the edge e of Q, and the two edges incident to v. The three lines carrying these segments define a line arrangement within R. This line arrangement can be subdivided into seven regions. Six of these regions come about by considering the region bounded by pairs of consecutive lines around the boundary of R. Let $B(t)$, $C(t)$, $D(t)$, $E(t)$, $F(t)$, and $G(t)$ denote the areas of these regions as a function of t. In Fig. 4(b)) these are indicated by the letters B through G. The seventh area ($H(t)$) is the area of the triangular region at the center of the arrangement. Observe that in Fig. 4(b), the regions representing the areas $B(t)$, $D(t)$, and $F(t)$ contain the central triangle whose area is given by $H(t)$. Because placements t and t' are combinatorially equivalent about the boundary of R, as the edge e is translated to its new placement at t', each of these area functions varies according to a single polynomial $I(t)$ of degree 2 in the coordinates of t. (This was argued in the proof of Lemma 3.2 because the difference in area is given by a trapezoid of linearly varying height). Also the area-of-overlap function varies as a single polynomial outside of R because the two placements are combinatorially equivalent outside of R. Thus it suffices to show that as the line segment (t, t') passes the point of criticality, the area-of-overlap function changes only in the addition or subtraction of the area functions of one of these seven regions. In fact, the change will be the addition or subtraction of the area function $H(t)$.

A complete proof of this would involve an exhaustive analysis of the possible configurations of the arrangements. Let us consider a couple to illustrate the idea. The other cases are similar. Suppose in Fig. 4(b), that the area of overlap consists of the region lying below the boundary of P and above the boundary of Q. Then the area of overlap, $\mathrm{Ar}(t)$ is equal to the sum

$$\mathrm{Ar}(t) = B(t) + F(t) - H(t) + I(t).$$

$H(t)$ is subtracted because both regions overlap $H(t)$, so its area is counted twice. As we move to the configuration shown in Fig. 4(c), the area of overlap, below P and above Q can now be seen to be

$$\mathrm{Ar}(t) = B(t) + F(t) + I(t)$$

(where the relevant regions are labeled B' and F' in the figure). Thus the incremental difference in the two area functions is the addition of $H(t)$.

As another example, suppose that in Fig. 4(b), the area of overlap consists of the region lying below P and below Q. Then the area of overlap is equal to

$$\mathrm{Ar}(t) = C(t) + D(t) + E(t) - H(t) + I(t).$$

(Again $H(t)$ is subtracted because $D(t)$ contains $H(t)$ but $H(t)$ is not part of the overlap.) As we move to the configuration in Fig. 4(c), the area below P and below Q is equal to

$$\mathrm{Ar}(t) = C(t) + D(t) + E(t) - 2H(t) + I(t)$$

(where the relevant regions are labeled C', D', etc. in the figure). The subtraction of $2H(t)$ is because $C(t)$ and $E(t)$ both contain $H(t)$, but it is not part of the overlap.) Thus the incremental difference in the two area functions is the subtraction of $H(t)$.

Since the area function $H(t)$ is a polynomial of degree 2 in the coordinates of the translation vector t, we can compute the incremental change by simply computing the coefficients for this polynomial, and then subtracting these from the current set of coefficients. In this way we can update the area-of-overlap function in $O(1)$ time per face visited. Since the total number of faces visited is $O(\mathrm{Arr}(P,Q)) = O((mn)^2)$ this is the amount of time needed to compute the area-of-overlap function. The running time is dominated by the $O(mn\log(mn) + \mathrm{Arr}(P,Q))$ time needed to compute the arrangement of critical placements. This establishes item (i) of the theorem.

To establish item (ii), observe that the space complexity is proportional to the size of the arrangement of critical placements, which is at most $O((mn)^2)$. For (iii), we can apply any standard point location algorithm to determine the face of the arrangement of critical placements containing t, and in constant time compute the area function at t for the resulting function. $\square$

5 Contour Tracing

Given simple polygons P and Q, and given an area value α, define the α-contour of P and Q to be the set of points t in the plane such that $\mathrm{Ar}(t) = \alpha$. One way to compute a contour is to apply the results of the previous section to compute the entire area function, and then compute the contour within each one of the pieces. In this section we consider the following problem. Given two polygons P and Q, and a single point t_0, trace the connected component of α-contour that contains t_0. (The questions of how to locate a point of a given value α and how to determine the set of all connected components of a given α value are both nontrivial, short of a complete traversal of the area function.) Observe that in general a contour can consist of multiple connected components, in fact as many as $\Omega((mn)^2)$ from the worst-case example given in the proof of Theorem 3.1. In certain degenerate cases each connected component of a contour may not consist of just a simple closed curve in the plane. For example, a contour may broaden into a 2-dimensional region if the area function (when viewed as a 3-dimensional surface) flattens into a plateau, or there may be degenerate points where multiple contour curves intersect each other. We will simplify the presentation by assuming that α is a general value so these degeneracies do not arise. With this assumption, from Corollary 3.1, the intersection of the α-contour with any cell in the arrangement of critical placements, $\mathrm{Crit}(P,Q)$, consists of a conic curve.

The area function at t_0 is determined by the face of $\mathrm{Crit}(P,Q)$ that contains this point. To avoid $O((mn)^2)$ time for computing the entire arrangement, we can compute the area function at t_0, in $O((m+n)\log(m+n)+mn)$ time, as we did in the previous section, by computing the single-placement arrangement at the point t_0, and building the area function from this arrangement through the method given in Lemma 3.2. Because the contour curve is a conic, we can apply standard methods from the theory of algebraic curves to convert it into a rational parametric form [8]. Thus, we can describe each point on the contour as a function $t(s)$, for a real parameter $s \geq 0$, where $t(0) = t_0$.

To trace the curve, we need to determine the parameter value s at which a translation vector traveling along the contour curve induces the next critical placement. Again, we do not want to take the time to compute the entire arrangement of critical placements, $\mathrm{Crit}(P,Q)$. Instead, we consider the placement (P, t_0+Q). As we translate Q along the contour, a transition in the area function occurs at the smallest positive parameter value s such that either (1) a vertex of $t(s)+Q$ intersects an edge of P, or (2) a vertex of P intersects an edge of $t(s)+Q$. To determine this value of s in case (1) we imagine shooting a bullet starting at each vertex v of Q, that travels along the contour $t(s)+v$ until it first strikes the boundary of P. See Fig. 5(a). In case (2) we shoot a bullet from each vertex v of P, that travels along the contour $-t(s)+v$ until it first strikes the boundary of Q (observing that this implies that the corresponding boundary point of $t(s)+Q$ intersects v). See Fig. 5(b). Among all $O(m+n)$ such bullets, we take the one that hits first, in the sense that the intersection point has the smallest value of s. (In Fig 5 this will be achieved by the lower right vertex of P.) This will be the smallest s such that the placement $(P, t(s)+Q)$ is critical.

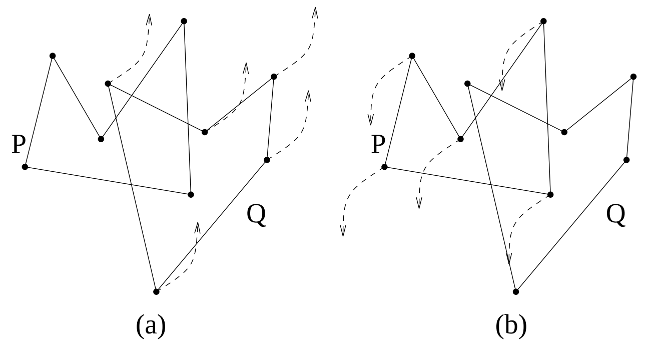

Figure 5: Determining the next critical placement by bullet shooting.

Unfortunately, we know of no data structure that will support bullet shooting along rational parametric functions of bounded algebraic degree in a time bound that is based on the combinatorial complexity of the objects. However, many standard ray shooting and ray tracing data algorithms [7, 13] can be generalized to handle tracing of parametric curves.

Once we have traced the contour to the next critical placement, we can apply the technique described in Lemma 4.1 to incrementally update the area-of-overlap function. Once this is done we can go back, derive the new contour, and continue the process. This process is repeated from one critical placement to the next, until returning to the initial configuration (which must occur because of our assumption that the contour is a simple closed curve). We lack theoretical bounds on the complexity of bullet shooting, and in general we cannot bound the number of transitions of critical placements the tracing algorithm will encounter. However, it is reasonable to believe that this approach is likely in many practical cases to be more efficient than computing the entire arrangement of critical placements. In summary, we have the following.

THEOREM 5.1 *Given two polygons P and Q, and a point t_0 on an α-contour of P and Q, we can trace the connected component of the contour containing t_0 in $O(mn + k(T(m) + T(n)))$ time, where m and n are the number of edges in the two polygons, k is the number of critical transitions visited by the contour trace, and $T(m)$ is the time to perform "ray-shooting" queries along curved rays represented by rational parametric conic curves, through a collection of m line segments.*

6 Concluding Remarks

The area-of-overlap function has interesting applications in many areas such as object recognition by probing. In this paper we considered the area of overlap of translated simple polygons. We showed that the area-of-overlap function of two simple polygons with m and n sides respectively is a continuous piecewise surface of polynomials of degree at most 2, which has up to $O((mn)^2)$ pieces. An efficient algorithm to compute the function was presented to compute a representation of the function in $O(mn \log(mn) + \text{Arr}(P, Q)) \leq O((mn)^2)$ time, where $\text{Arr}(P, Q)$ is the complexity of the arrangement of critical placements of P and Q. The representation of the function has space complexity $O(\text{Arr}(P, Q)) \leq O((mn)^2)$. We also presented a technique for tracing a single contour curve without explicitly constructing the entire critical arrangement. Some interesting related questions are:

- What if we restrict the polygons to be convex polygons? An important special case is computing the area-of-overlap function for a convex polygon with itself.
- Can the results of Section 5 be extended to find *all* contour lines of fixed area in an efficient manner (without computing the entire area function)?
- Is it possible to find the translation that maximizes the area of overlap (without computing the entire function)?
- What if the sides of the polygons are curves (e.g. low-degree splines) rather than straight lines?
- What if rotations are allowed?

References

[1] E. Arkin, H. Meijer, J. Mitchell, D. Rappaport, and S. Skiena. Decision trees for geometric models. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 369–378, 1993.

[2] E. M. Arkin, M. T. Goodrich, J. S. B. Mitchell, D. Mount, C. D. Piatko, and S. S. Skiena. Point probe decision tres for geometric concept classes. In *Proc. 3nd Workshop Algorithms Data Struct.*, volume 709 of *Lecture Notes in Computer Science*, pages 95–106. Springer-Verlag, Berlin, 1993.

[3] B. Chazelle and H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *J. ACM*, 39:1–54, 1992.

[4] R. Cole and C. K. Yap. Shape from probing. *J. Algorithms*, 8:19–38, 1987.

[5] D. P. Dobkin, H. Edelsbrunner, and C. K. Yap. Probing convex polytopes. In *Proc. 18th Annu. ACM Sympos. Theory Comput.*, pages 424–432, 1986.

[6] H. Edelsbrunner and S. S. Skiena. Probing convex polygons with x-rays. *SIAM J. Comput.*, 17:870–882, 1988.

[7] J. Hershberger and S. Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. In *Proc. 4th ACM-SIAM Sympos. Discrete Algorithms*, pages 54–63, 1993.

[8] C. Hoffmann. *Geometric and Solid Modeling.* Morgan Kaufmann, San Mateo, CA, 1989.

[9] K. Kedem, R. Livne, J. Pach, and M. Sharir. On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete Comput. Geom.*, 1:59–71, 1986.

[10] D. M. Mount. The densest double-lattice packing of a convex polygon. In J. E. Goodman, R. Pollack, and W. Steiger, editors, *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, pages 245–262. AMS, Providence, RI, 1991.

[11] D. M. Mount and R. Silverman. Packing and covering the plane with translates of a convex polygon. *J. Algorithms*, 11:564–580, 1990.

[12] R. Pollack, M. Sharir, and S. Sifrony. Separating two simple polygons by a sequence of translations. *Discrete Comput. Geom.*, 3:123–136, 1987.

[13] H. Samet. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley, Reading, MA, 1990.

[14] S. S. Skiena. Problems in geometric probing. *Algorithmica*, 4:599–605, 1989.

[15] S. S. Skiena. Probing convex polygons with half-planes. *J. Algorithms*, 12:359–374, 1991.

[16] S. S. Skiena. Interactive reconstruction via geometric probing. *Proc. IEEE*, 80:1364–1383, 1992.