Computing the Maximum Overlap of Two Convex Polygons Under Translations

Mark De Berg, Olivier Devillers, Marc Van Kreveld, Otfried Schwarzkopf, Monique Teillaud

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Computing the Maximum Overlap of Two Convex Polygons Under Translations

Mark de Berg, Olivier Devillers, Marc van Kreveld, Otfried Schwarzkopf,

Monique Teillaud

N 2832

Mars 1996

THÈME 2

Rapport de recherche

ISSN 0249-6399

Computing the Maximum Overlap of Two Convex Polygons Under Translations

Mark de Berg, Olivier Devillers, Marc van Kreveld, Otfried Schwarzkopf, Monique Teillaud

Thème 2 — Génie logiciel
et calcul symbolique
Projet Prisme

Rapport de recherche n2832 — Mars 1996 — 22 pages

Abstract: Let P be a convex polygon in the plane with n vertices and let Q be a convex polygon with m vertices. We prove that the maximum number of combinatorially distinct placements of Q with respect to P under translations is $O(n^2 + m^2 + \min(nm^2 + n^2m))$, and we give an example showing that this bound is tight in the worst case. Second, we present an $O((n+m)\log(n+m))$ algorithm for determining a translation of Q that maximizes the area of overlap of P and Q.

We also prove that the position which translates the centroid of Q on the centroid of P always realizes an overlap of $9/25$ of the maximum overlap and that this overlap may be as small as $4/9$ of the maximum.

Key-words: computational geometry, arrangement, localisation

(Résumé : tsvp)

This work was supported by ESPRIT Basic Research Action No. 7141 (project ALCOM II: *Algorithms and Complexity*). M.d.B. was also supported by the Dutch Organisation for Scientific Research (N.W.O.).

Dept. of Computer Science, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, the Netherlands.

INRIA, B.P.93, 06902 Sophia-Antipolis cedex (France),

Dept. of Computer Science, Utrecht University.

Dept. of Computer Science, Utrecht University. Present address: Dept. of Computer Science, Postech, San 31, Hyoja-dong, Pohang, Kyungbuk 790-784, South Korea.

INRIA,

Placement avec recouvrement maximal de deux polygones convexes en translations

Résumé : Soit P et Q deux polygones convexes à n et m sommets se deplaçant en translation dans le plan. Nous montrons que le nombre de positions combinatoirement différentes de P par rapport à Q est borné par $O(n^2 + m^2 + \min(nm^2 + n^2m))$ et nous exhibons un exemple prouvant que cette borne est optimale dans le cas le pire. Nous présentons une algorithme calculant la translation maximisant l'aire de l'intersection des deux polygones en temps $O((n + m)\log(n + m))/$

Nous prouvons également que la translation qui fait coïncider les centres de gravité de P et Q permet d'obtenir une aire de l'intersection d'au moins $9/25$ de l'aire maximale réalisable. Nous presentons un exemple ou l'aire de l'intersection n'est que de $4/9$ de l'optimale.

Mots-clé : géométrie algorithmique, arrangement, localisation

1 Introduction

Matching plays an important role in areas such as computer vision. Typically one is given two 'shapes'—point sets or polygons, for instance—and one wants to determine how much these shapes resemble each other. More precisely, one wants to find a rigid motion of one shape that maximizes the resemblance with the other shape. There are several ways to measure resemblance. For example, for point sets or polygonal chains one can use Hausdorff distance [ABB91, AST92, CGH$^+$93, HKK92, HKS93]; for polygonal chains one can also use Fréchet distance [AG92].

We consider the matching problem for convex polygons in the plane, and the rigid motions that we allow are translations. The resemblance of two convex polygons can be measured by looking at the Hausdorff or Fréchet distance between their boundaries, but it seems more appropriate to look at the area of overlap of the two polygons. Notice that maximizing the area of overlap of two polygons is equivalent to minimizing the area of the symmetric difference. One of the advantages of measuring resemblance by looking at the area of overlap instead of at the Hausdorff distance between the boundaries is that the area of overlap is less sensitive to noise in the image: noise may add thin features to the boundary but is unlikely to add large areas. The problem of measuring resemblance of polygons by looking at their area of intersection has been looked at by Mount *et al.* [MSW93] who pose as an open problem the case of two convex polygons. An algorithm with $O(n(n+m))$ time complexity is known for finding the maximum overlap area for two convex polygons, one of which is allowed to rotate with one point on its boundary sliding on the other polygon's boundary [Ven95].

To compute the resemblance of two polygons under translations where the measure is the area of overlap, we have to solve the following problem: given two polygons P and Q in the plane, find a translation of Q that maximizes the area of overlap with P. Our results are as follows. Let n and m denote the number of vertices of P and Q, respectively. We start by studying a combinatorial question: how many combinatorially distinct placements of Q with respect to P are there? Here we define two placements to be combinatorially equivalent if the same pairs of edges (one from P and one from Q) intersect—see Section 2 for a more precise definition. We show that the number of distinct placements is $O(n^2 + m^2 + \min(nm^2 + n^2m))$, and we give an example showing that this bound is tight in the worst case. To our surprise, this result appears to be new: previous work on bounding the number of placements of a polygon in a polygonal environment is usually motivated by motion planning problem and, hence, only deals with the case where the polygon is not allowed to intersect the environment—see Latombe's book [Lat91] or Halperin's thesis [Hal92] and the references in those works. Our main result is presented in Section 3, where we give an $O((n+m)\log(n+m))$ time algorithm for computing a placement of Q that maximizes the area of overlap with P. Our algorithm is based on a theorem stating that the area-of-overlap function is unimodal; this result is of independent interest. In a last section, we study the particular placement where the centroids of the two polygons coincide. We prove that the ratio between the areas of overlap realized by this placement and the optimal placement is bounded.

Our work can also be seen as a generalization of the problem of placing a copy of one polygon inside another polygon. Chazelle [Cha83] studied several variants of this problem. One of his results is that, given two convex polygons P and Q, one can decide in linear time whether Q can be translated such that it is contained in P. Other papers compute the largest copy of a polygon that can be placed inside another one [AB88, CK93, For85, SCK$^+$86].

2 The number of distinct placements

Let P be a simple polygon with n vertices in the plane and let Q be a simple polygon with m vertices. The position and orientation of P are fixed, but Q is free to translate. In this section we bound the number of distinct placements of Q with respect to P. We first define formally when we call two placements distinct.

We denote the boundary of P by ∂P, and the boundary of Q by ∂Q. We consider boundary edges to be relatively open sets, that is, their endpoints are not included. Let r_Q be a reference point on Q, say the lexicographically smallest vertex. For a point r in the plane, $Q(r)$ denotes Q with its reference point placed at r. Similarly, for an edge e or a vertex v of Q, $e(r)$ and $v(r)$ denote the edge e and vertex v when Q is placed at r. We call $Q(r)$ a *placement* of Q. The space of all possible placements of Q—in our case this is a 2-dimensional space–is called the *configuration space* [Lat91].

Definition 2.1 *The* intersection set *of P and a placement $Q(r)$, denoted $I(r)$, is the set consisting of all pairs (f,g) such that f is the interior of P, an edge of P, or a vertex of P, g is the interior of $Q(r)$, an edge of $Q(r)$, or a vertex of $Q(r)$, and f and g intersect. Two placements $Q(r)$ and $Q(r')$ are* combinatorially distinct *if and only if $I(r) \neq I(r')$.*

The configuration space can be partitioned into regions according to the intersection sets of the corresponding placements: two points are in the same region if and only if the corresponding placements are combinatorially equivalent. Hence, bounding the the number of combinatorially distinct placements boils down to studying the configuration space.

Previous work on configuration spaces was usually inspired by motion planning applications. In this application the polygon Q is a robot, the polygon P is an obstacle, and one is interested in the part of the configuration space where Q does not collide with P or, in other words, the region where the intersection set is empty. This region of configuration space is called the *free space*, and placements in it are called *free placements*. When only translations are considered, then the free space is the complement of the the Minkowski sum of P and $-Q$, and its complexity is $\Theta(n^2m^2)$ in the worst case; for convex polygons, the complexity is $\Theta(n+m)$ in the worst case. For more information and references on configuration spaces in connection with motion planning we refer the reader to Latombe's book [Lat91] or Halperin's thesis [Hal92]. In our vision application we are also interested in placements where the intersection set is not empty, so few results from the motion planning literature apply [PT92].

Let's have a closer look at the configuration space. Fix an edge e of P and an edge e' of Q, and consider the locus of all points r such that e intersects $e'(r)$. This region is a parallelogram, denoted $\pi(e,e')$, spanned by a translated copy of e and a translated copy of e'. Observe that for points r in the interior of the edges of $\pi(e,e')$, a vertex of e lies on $e'(r)$ or a vertex of $e'(r)$ lies on e; for a point r that is a vertex of $\pi(e,e')$, a vertex of e coincides with a vertex of $e'(r)$. Let

$$\Pi = \{\pi(e,e') : \ e \text{ is an edge of } P,\ e' \text{ is an edge of } Q\}.$$

The arrangement $\mathcal{A}(\Pi)$ induced by Π is the partitioning of configuration space we mentioned above: there is a one-to-one correspondence between the combinatorially distinct placements and the faces, arcs,[1] and nodes of $\mathcal{A}(\Pi)$. So a bound on the complexity of $\mathcal{A}(\Pi)$ immediately implies a bound on the number of distinct placements.

We proceed to bound the complexity of $\mathcal{A}(\Pi)$. Because $\mathcal{A}(\Pi)$ is a planar subdivision defined by nm parallelograms, its complexity is bounded by $O(n^2m^2)$. For simple polygons, this bound is tight in the worst case. In fact, the complexity of the the cells where the intersection set is empty can already be $\Theta(n^2m^2)$, as observed above. The lower bound example that achieves $\Theta(n^2m^2)$ complexity uses non-convex polygons. One would expect that the maximum number of combinatorially distinct placements of two convex polygons is significantly less than $\Theta(n^2m^2)$. This is indeed

[1] To avoid confusion between the edges of the polygons and the edges of $\mathcal{A}(\Pi)$, we call the latter arcs. Similarly, we call the vertices of $\mathcal{A}(\Pi)$ nodes.

the case: the next theorem states that this number is $\Theta(n^2 + m^2 + \min(nm^2, n^2m))$. Notice that, unlike for simple polygons, the number of distinct placements of two convex polygons is much larger if we consider all placements than if we consider free placements only.

Theorem 2.2 *The maximum number of combinatorially distinct placements of two convex polygons with n and m vertices, respectively, is:*

$$\Theta(n^2 + m^2 + \min(nm^2, n^2m)).$$

Proof: We first prove the upper bound. Let P be a convex polygon with n vertices, and Q a convex polygon with m vertices. We bound the complexity of the subdivision $\mathcal{A}(\Pi)$ we get for P and Q, as defined above. Because $\mathcal{A}(\Pi)$ is a planar subdivision, it suffices to bound its number of nodes. A node of $\mathcal{A}(\Pi)$ is either a corner of some parallelogram $\pi(e, e')$ or an intersection between the boundary of two such parallelograms. The corners of the parallelograms corresponds to a placement where a vertex of Q coincides with a vertex of P; clearly there are $O(nm)$ of these placements. The intersections between parallelogram boundaries correspond to placements such that

i. there are edges e_1, e_2 of P and vertices v_1, v_2 of Q such that $v_i \in e_i$, for $i = 1, 2$, or

ii. there are vertices v_1, v_2 of P and edges e_1, e_2 of Q such that $v_i \in e_i$, for $i = 1, 2$, or

iii. there is an edge e_1 of P, a vertex v_2 of P, a vertex v_1 of Q and an edge e_2 of Q such that $v_i \in e_i$, for $i = 1, 2$.

First we bound the number of nodes of type (i). Fix one vertex v_1 of Q, and place v_1 somewhere on ∂P. Now move Q 'around' P, while keeping v_1 on ∂P. We get a type (i) node when a vertex of Q crosses a edge of P. Because the path that every vertex of Q describes is a translate of ∂P, it can intersect ∂P at most twice. Hence, the total number of type (i) nodes involving vertex v_1 is at most $2m$. The total number of type (i) nodes over all vertices of Q is therefore $O(m^2)$.

A similar argument shows that the number of type (ii) nodes is $O(n^2)$.

It remains to bound the number of nodes of type (iii). We fix a vertex v_2 of P and move Q 'around' P while v_2 stays on ∂Q. We must count the number of times that a vertex of Q crosses an edge of P. Let's look at the path that a vertex v_1 of Q follows. This path can be obtained by placing v_1 at v_2 and rotating Q over 180 degrees around v_1; the mirrored image of Q that results is exactly the path that v_1 follows. So the path is convex and polygonal, and it has m segments. Hence, v_1 crosses ∂P at most $O(\min(n,m))$ times. The number of type (iii) nodes involving vertex v_2 of P is therefore $O(m\min(n,m))$, and the total number of type (iii) nodes is $O(nm\min(n,m))$.

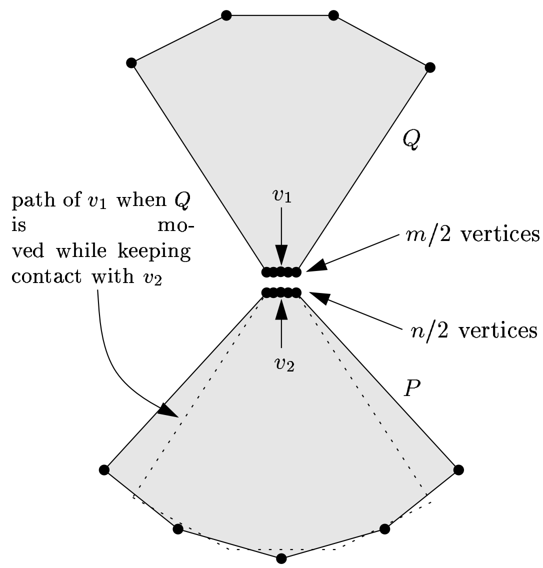

Figure 1: Two convex polygons with $\Omega(\min(nm^2+n^2m))$ distinct placements.

An example where there are $\Omega(n^2+m^2)$ distinct placements is easy to construct, so we only give an example with $\Omega(nm\min(n,m))$ distinct placements. Fig. 1 gives such an example. The dotted polygonal closed path is the path v_1 follows when Q is moved while keeping contact with v_2. This path intersects ∂P $\Omega(\min(n,m))$ times. Let w_1 be any of the $m/2$ bottom vertices of Q and w_2 any of the $n/2$ top vertices of P. When the top vertices of P and the bottom vertices of Q are placed close enough together, then the path followed by w_1 when Q is moved around w_2 will be close enough to the dotted path, so that there will be $\Omega(\min(n,m))$ intersections of the path with ∂P. Hence, we get a total of $\Omega(nm\min(n,m))$ distinct placements. $\square$

3 Computing the maximum overlap

We now get to the main problem studied in this paper: given two convex polygons P and Q, find a placement of Q that maximizes the overlap with P. First, we need to introduce some notation. The *overlap function* $\omega(r) : \mathbb{R}^2 \to \mathbb{R}$ of P and Q is defined as

$$\omega(r) := \text{ the area of } P \cap Q(r).$$

Our problem is thus to find a placement $Q(r)$ that maximizes $\omega(r)$. We call such a placement a *goal placement.*

We first look at a restricted version of the problem, where Q is only allowed to translated into a fixed direction. Without loss of generality, we assume this direction is horizontal. Thus, for a given value y^*, we define the *(horizontal) overlap function at* y^*, denoted by $\omega_{y^*}(t)$, as

$$\omega_{y^*}(t) := \omega((t, y^*)).$$

A non-negative function $\chi : D \to \mathbb{R}$ is called *unimodal* if there is an interval $D = [a_0 : a_1]$ and points $b_0, b_1 \in D$ with $b_0 \leqslant b_1$ such that χ is zero outside D, strictly increasing from a_0 to b_0, constant from b_0 to b_1, and strictly decreasing from b_1 to a_1. Our algorithm is based on the unimodality of the horizontal overlap function.

Theorem 3.1 *Let P and Q be two convex polygons then the square root $\sqrt{\omega}$ of the bimodal overlap function is a concave function.*

Proof: We will prove that the monovariate fonction $\sqrt{\omega_{y^*}(t)}$ is concave, and then the claim follows since a function such that all cross sections are concave is concave. Imagine moving Q from left to right over the plane, starting with $Q((-\infty, y^*))$ and ending at $Q((+\infty, y^*))$. Define $Q(t) := Q((t, y^*))$, and $A(t) := P \cap Q(t)$. Thus $A(t)$ is the intersection of P and Q at time t. We define a three-dimensional polytope $\mathcal{P}_{PQ}$ by viewing time as the third dimension, and taking the union of all polygons $A(t)$:

$$\mathcal{P}_{PQ} := \{(x, y, t) : (x, y) \in A(t)\}.$$

The cross-section of $\mathcal{P}_{PQ}$ with the horizontal plane $t = t^*$ is exactly the intersection $A(t^*)$. Thus our problem can be phrased as follows: given the three-dimensional polytope $\mathcal{P}_{PQ}$, is the square root of the function that describes the area of intersection of $\mathcal{P}_{PQ}$ and a horizontal plane h convex, as we sweep h through $\mathcal{P}_{PQ}$? As Avis *et al.*[ABS$^+$96] we remark that a direct application of the Brunn-Minkowski theorem [Grü67] allows to conclude that this is indeed the case if $\mathcal{P}_{PQ}$ is convex.

To prove that $\mathcal{P}_{PQ}$ is convex, it suffices to write it as the intersection of two convex polytopes

$$\mathcal{P}_{PQ} = \{(x, y, t) : (x, y) \in P\} \cap \{(x, y, t) : (x, y) \in Q(t)\}$$

$\square$

Theorem 3.2 *Let P and Q be two convex polygons, and let $y^* \in \mathbb{R}$. Then horizontal overlap function $\omega_{y^*}(t)$ is unimodal.*

Proof: As in Avis *et al.*[ABS+96] this theorem is a straightforward consequance of Theorem3.1. $\square$

The area of overlap behaves as follows: after being zero for some time it starts increasing until it reaches a maximum, then it may stay constant for some time, until it starts to decrease again and eventually becomes (and stays) zero.

Theorem 3.2 can be used to compute the maximum overlap of P and Q for the case where Q is confined to translate along a fixed line. This algorithm will be an important ingredient of the general algorithm.

Lemma 3.3 *[Avis* et al.*] For a line ℓ we can compute $\max_{r\in\ell} \omega(r)$ in $O(n+m)$ time.*

Proof: Using Chazelle's algorithm[Cha92] the convex polytope $\mathcal{P}_{PQ}$ can be computed in linear time, and then Avis *at al.* algorithm [ABS+96] compute the horizontal section of $\mathcal{P}_{PQ}$ of maximal area in linear time. $\square$

We now turn our attention to the general case, where arbitrary translations are allowed. Our algorithm consists of two stages. In the first stage we locate a horizontal strip that contains the reference point of a goal placement. This will be done by a binary search that uses the algorithm from Lemma 3.3 as a subroutine. This reduces the complexity of the search space sufficiently to enter the second stage of the algorithm, which is based on cuttings. The second stage reduces the complexity of the search space further so that it becomes easy to compute the maximum overlap. We now describe the stages in more detail.

The first stage. Consider a placement where Q is completely below P, and imagine moving Q upward until it is entirely above P. Let $Y = y_1, y_2, \ldots, y_{nm}$ be the sorted sequence of y-values where a vertex of Q and a vertex of P align horizontally. In other words, Y contains the values y_i such that there are vertices v of P and w of

$Q((x, y_i))$ with the same y-coordinate. We shall do a binary search on Y to locate a horizontal strip $[-\infty : \infty] \times [y_i : y_{i+1}]$ that contains a goal placement. (In fact, we should write 'that contains the reference point of a goal placement'. When no confusion can arise, we shall permit ourselves this slight abuse of terminology.) We do not compute the set Y explicitly, however, because Y can contain nm elements and we do not want to spend that much time.

Let's look more closely at the set Y. Let $A = \{a_1, \ldots, a_n\}$ be the set of y-coordinates of the vertices of P, sorted in increasing order, and let $B = \{b_1, \ldots, b_n\}$ be the set of y-coordinates of the vertices of $Q((0,0))$, sorted in decreasing order. The sets A and B can be computed in linear time. The elements of the set Y are exactly the entries of the matrix

$$\mathcal{M} = (c_{ij}), \text{ where } c_{ij} = a_i - b_j.$$

Because the sets A and B are sorted, every row and every column of $\mathcal{M}$ is sorted. Furthermore, an entry c_{ij} can be evaluated in constant time. Hence, for any parameter k with $1 \leqslant k \leqslant nm$, we can compute the k-th largest entry of $\mathcal{M}$ in $O(m \log(2n/m))$ time with an algorithm by Frederickson and Johnson [FJ84]. In fact we will use this complexity in the less powerfull form $O(n+m)$ which is convenient for our purpose, is $n \leqslant m$ $m\log(2n/m) = O(m)$ and if $n > m$ $m\log(2n/m) < 2n \max_{2 \leqslant x} \frac{\log x}{x} = O(n)$.

The binary search now proceeds as follows. In a generic step we have two values, $k_{\min}$ and $k_{\max}$ such that there is a goal placement in the horizontal strip$[-\infty : \infty] \times [y_{k_{\min}} : y_{k_{\max}}]$. Initially $k_{\min} = 0$ and $k_{\max} = nm$. We first compute the values y_k and y_{k+1}, where $k = \lfloor (k_{\min} + k_{\max})/2 \rfloor$, with the algorithm of Frederickson and Johnson. Then we compute $\max_t \omega_{y_k}(t)$ and $\max_t \omega_{y_{k+1}}(t)$ using Lemma 3.3. There are three cases to consider, depending on the computed values:

If $\max_t \omega_{y_k}(t) < \max_t \omega_{y_{k+1}}(t)$ then we set $k_{\min} := k$.

If $\max_t \omega_{y_k}(t) > \max_t \omega_{y_{k+1}}(t)$ then we set $k_{\max} := k+1$.

If $\max_t \omega_{y_k}(t) = \max_t \omega_{y_{k+1}}(t)$ then we set $k_{\min} := k$ and $k_{\max} := k+1$ and we have found the strip.

The binary search continues until either a goal placement has been found, or $k_{\max} = k_{\min} - 1$. The correctness of the algorithm is based on the following lemma.

Lemma 3.4 *Let ℓ_1 and ℓ_2 be two lines, and let r_1 and r_2 be points on ℓ_1 and ℓ_2, respectively, such that $\omega(r_1) = \max_{r \in \ell_1} \omega(r)$ and $\omega(r_2) = \max_{r \in \ell_2} \omega(r)$. If $\omega(r_1) \geqslant \omega(r_2) > 0$ and r_1 does not lie on ℓ_2 then the open half-plane bounded by ℓ_2 and containing r_1 contains a goal placement.*

Proof: We shall prove that the closed half-plane bounded by ℓ_2 and not containing

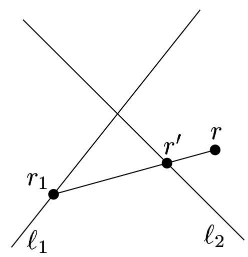

Figure 2:

r_1 cannot contain a placement r such that $\omega(r) > \omega(r_1)$, which implies the lemma. Let r be any point in this half-plane, and let r' be the intersection point of the closed line segment $\overline{rr_1}$ with ℓ_2. Because the overlap function is unimodal, $\omega(r) > \omega(r_1)$ would imply $\omega(r') > \omega(r_1)$. But since $\omega(r_2) \geqslant \omega(r')$ by definition of r_2, this would contradict the assumption that $\omega(r_1) \geqslant \omega(r_2)$. □

We can now prove that the binary search algorithm correctly and efficiently finds a goal placement.

Lemma 3.5 *The binary search finds in time $O((n+m)\log(n+m))$ a horizontal strip:*

$$[-\infty : \infty] \times [y_i : y_{i+1}]$$

that contains a goal placement.

Proof: Let's first prove that the algorithm is correct. This amounts to proving that the three cases mentioned above are handled correctly. Let t_1 be a value maximizing $\omega_{y_k}(t)$ and let t_2 be a value maximizing $\omega_{y_{k+1}}(t)$. Define $r_1 = (t_1, y_k)$ and $r_2 = (t_2, y_{k+1})$. Suppose that $\omega(r_1) \leqslant \omega(r_2)$. By Lemma 3.4 there must be a goal placement above the line $y = y_k$, which proves that the first case is handled correctly. Similarly, $\omega(r_1) \geqslant \omega(r_2)$ implies that there is a goal placement below the line $y = y_{k+1}$, which proves that the second case is handled correctly. By combining the arguments for the first two cases, we see that also the third case is handled correctly.

It remains to prove the time bound. In each step of the binary search we use the selection algorithm of Frederickson and Johnson [FJ84], which takes $O(m+n)$ time, and we apply the algorithm of Lemma 3.3, which takes $O(n+m)$ time. Since the number of steps of the binary search is $O(\log(nm))$, the total time is as claimed. □

The binary search on the set Y gives us a horizontal strip that contains a goal placement. For any placement $Q(r)$ in the interior of this strip, the vertical order of the vertices of P with respect to those of $Q(r)$ is fixed. This means that the complexity of the part of $\mathcal{A}(\Pi)$ within R is significantly less than the total complexity of $\mathcal{A}(\Pi)$, as we show next.

Lemma 3.6 *After the first stage of the algorithm we have located a horizontal strip $\sigma = [-\infty : \infty] \times [y : y']$ containing a goal placement such that the part of $\mathcal{A}(\Pi)$ inside σ is formed by $O(n+m)$ segments.*

Proof: Recall that $\mathcal{A}(\Pi)$ is defined by $O(nm)$ parallelograms. Each parallelogram is defined by a pair of edges, one from P and one from Q. The edges of these parallelograms, in other words, the segments that induce $\mathcal{A}(\Pi)$, are defined by a vertex-edge pair. We claim that a vertex can define at most two vertex-edge pairs whose corresponding segment intersects σ. Let v be a vertex of Q, and let e be an edge of P. Let $Q(r)$ be a placement with $r \in \sigma$. If the horizontal line through $v(r)$ does not intersect e then v exchanges its vertical order with an endpoint of e when it is moved to lie on e. Hence, v can only define a vertex-edge pair with an edge e intersected by the horizontal line through $v(r)$. Because P is convex there are at most two such edges. The same argument shows that any vertex of P can define at most two vertex-edge pairs. $\square$

The second stage We enter the second stage with a horizontal strip $\sigma = [-\infty : \infty] \times [y : y']$ that contains a goal position. The number of segments defining $\mathcal{A}(\Pi)$ inside σ is $O(n+m)$. From the proof of Lemma 3.6 it follows that we can compute these segments in linear time: take a point r inside the strip, and merge the two sorted sequences of y-coordinates of the vertices of P and the vertices of $Q(r)$ to find for each vertex the at most two edges with which it can define an edge inside σ. Because we know the segments defining $\mathcal{A}(\pi) \cap \sigma$, we can use cuttings to zoom in further on a goal placement. How this works is explained next.

Let S be a set of line segments in the plane. A $(1/k)$-*cutting* $\Xi(S)$ *for* S is a collection of triangles with disjoint interiors that collectively cover the entire plane, such that for each triangle in $\Xi(S)$ the number of segments intersecting its interior is at most $|S|/k$. The size of a cutting is the number of simplices it consists of. For any set of lines in the plane—and, hence, for any set of segments—there is a cutting of size $O(k^2)$. For constant k such a cutting can be constructed in linear time [Cha93].

Let $S(\sigma)$ be the set of segments defining $\mathcal{A}(\Pi)$ inside σ. We construct a (1/4)-cutting $\Xi(S(\sigma))$. This cutting consists of $O(1)$ triangles, each intersected by $|S(\sigma)|/4$ segments. The idea is to find a triangle in $\Xi(S(\sigma))$ that contains a goal placement, and to proceed recursively inside that triangle. (Actually, we will recurse in two triangles.) To decide in which triangle to recurse we proceed as follows.

Let $L = \{\ell_1, \ldots, \ell_a\}$ be the set of lines through the edges of the cutting $\Xi(S(\sigma))$. On each line ℓ_i we compute the maximum overlap $\xi_i = \max_{r \in \ell_i} \omega(r)$ in $O(n+m)$ time using the algorithm of Lemma 3.3. Let's assume for the moment that all the maxima are distinct. Let i^* be such that $\xi_{i^*} = \max_i \xi_i$. By Lemma 3.4 we know for each line ℓ_i with $i \neq i^*$ to which side we can restrict our attention. This implies that we can restrict our attention to at most two triangles (separated by the line ℓ_{i^*}). The number of segments on which we must recurse is thus at most $|S(\sigma)|/2$. After $O(\log(n+m))$ recursive calls we are left with two triangular regions that are not intersected by any of the segments of $\mathcal{A}(\Pi)$. This means that the overlap function is continuous inside each of these regions. In fact, it is a second-degree polynomial, which can be computed in linear time. Once we have the polynomial we can compute its maximum in constant time, giving us the desired goal placement. The total running time for the second stage is $O((n+m)\log(n+m))$. This almost finishes the description of the algorithm. It only remains to get rid of the assumption that all maxima ξ_i are distinct; this is done as follows.

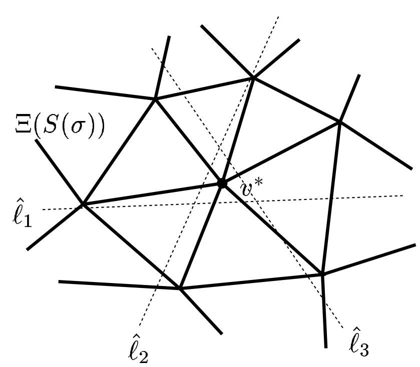

Figure 3: Dealing with a degenerate situation.

Let $\xi^* = \max_i \xi_i$. The difficulty arises when ξ^* is achieved at a vertex v^* of the cutting $\Xi(S(\sigma))$, as in Fig. 3. In this case there must be a goal position in one of the triangles of the cutting incident to v^*, but we do not know which one yet. If v^* itself is a goal position then it doesn't matter where we recurse (provided we keep track of the the placement with the largest overlap found so far), so let's assume that this is not the case. Now, to find a triangle containing a goal placement we take three lines $\hat{\ell}_1$, $\hat{\ell}_2$, and $\hat{\ell}_3$ such that v^* lies in the triangle $\Delta(\hat{\ell}_1, \hat{\ell}_2, \hat{\ell}_3)$ enclosed by them—see Fig. 3. The distance δ from v^* to each of the three lines should be such that there is no goal placement inside Δ. This can be achieved by computing with δ symbolically, treating it as an extension of the reals which is larger than zero but smaller than any positive real. Let $\hat{\xi}_j := \max_{r \in \hat{\ell}_j} \omega(r)$. We compute $\hat{\xi}_j$, for $j = 1, 2, 3$, using Lemma 3.3 Let $\hat{r}_j$ be such that $\omega(\hat{r}_j) = \hat{\xi}_j$.

Lemma 3.7 *If v^* itself is not a goal position, then $\hat{\xi}_j > \xi^*$ for at least one $j \in \{1, 2, 3\}$. Moreover, for such a j the triangle of $\Xi(S(\sigma))$ incident to v^* containing $\hat{r}_j$ must contain a goal placement.*

Proof: Follows from Lemma 3.4. □

Thus we can also find out where to recurse in $O((n+m)\log(n+m))$ time in degenerate cases. This completes the proof of our main result, which is summarized in the following theorem.

Theorem 3.8 *Let P be a convex polygon in the plane with n vertices, and let Q be a convex polygon with m vertices. Then a placement of Q that maximizes the area of $P \cap Q$ can be computed in $O((n + m)\log(n + m))$ time.*

4 Bounds on the overlap for a particular translation

We prove in this section that a position with good overlap can be found easily. More precisely, if the polygon Q is translated to a placement $Q(r)$ such that the centroid of $Q(r)$ coincides with the centroid of P, then the area of $P \cap Q(r)$ is at least $9/25$ times the maximal overlap area.

4.1 Lower bound

Let us first define some notations. The centroid of P is denoted by c_P.

$$\iint_{u \in P} u du \Big/ \iint_{u \in P} du = \iint_{u \in P} u du \,/ area(P)$$

Similarly c_Q denotes the centroid of Q.
In this section we will choose the origin O so that the overlap function is maximal at

the origin, that is the reference position $Q(O)$ for Q is a maximal overlap position. The maximal overlap area is thus denoted $\omega(O)$. In the sequel, we will use the polar coordinates (r, θ) with respect to that origin, and the horizontal direction; The point with polar coordinates $(1, \theta)$ will be denoted as e_θ.

We denote by Ω the three dimensional object bounded above by the graph of ω and below by the horizontal plane $\mathbb{R}^2$.

$$\Omega = \{(x, y, z) \in \mathbb{R}^3 : z = \omega(x, y)\}$$

Lemma 4.1 *The translation r that superimposes the centroids of P and $Q(r)$ is given by the projection of the centroid of Ω onto $\mathbb{R}^2$.*

Proof: The horizontal projection $p(c_\Omega)$ of the centroid c_Ω of Ω is

$$p(c_\Omega) = \iint_{v\in\mathbb{R}^2} \omega(v) v dv \Big/ \iint_{v\in\mathbb{R}^2} \omega(v) dv$$

replacing ω by its expression by integrals, we get

$$\begin{aligned}
&p(c_\Omega)\\
&= \iint_{v\in\mathbb{R}^2} \iint_{u\in P\cap(Q+v)} du\ v dv \Big/ \iint_{v\in\mathbb{R}^2} \iint_{u\in P\cap(Q+v)} du\ dv\\
&= \iiiint_{u\in P, v\in u-Q} v dv du \Big/ \iiiint_{u\in P, v\in u-Q} dv du\\
&= \iint_{u\in P} \iint_{v\in Q} -(u-v) dv\ du \Big/ \iint_{u\in P} \iint_{v\in Q} -dv\ du
\end{aligned}$$

then, using the definitions of c_P and c_Q, we obtain

$$\begin{aligned}
&p(c_\Omega)\\
&= \iint_{u\in P} (-u + c_Q) area(Q) du \Big/ \iint_{u\in P} -area(Q) du\\
&= \left(c_Q area(P) area(Q)) + area(Q) \iint_{u\in P} -u du\right)\\
&\qquad \times \frac{1}{-area(P) area(Q)}\\
&= c_P - c_Q
\end{aligned}$$

□

$p(c_\Omega)$ can now be evaluated in polar coordinates:

$$
\begin{aligned}
&p(c_\Omega)\\
&= \iint_{v\in\mathbb{R}^2} \omega(v)v dv \Big/ \iint_{v\in\mathbb{R}^2} \omega(v)dv\\
&= \int_{\theta=0}^{2\pi}\int_{r=0}^{\infty} \omega(re_\theta)re_\theta r dr d\theta \Big/ \int_{\theta=0}^{2\pi}\int_{r=0}^{\infty} \omega(re_\theta)r dr d\theta\\
&= \int_{\theta=0}^{2\pi}\int_{r=0}^{\infty} \omega(re_\theta)r^2 dr\ e_\theta d\theta \Big/ \int_{\theta=0}^{2\pi}\int_{r=0}^{\infty} \omega(re_\theta)r dr\ e_\theta d\theta\\
&= \int_{\theta=0}^{2\pi} e_\theta \rho(\theta)A(\theta)d\theta \Big/ \int_{\theta=0}^{2\pi} e_\theta A(\theta)d\theta
\end{aligned}
\tag{1}
$$

Where we define $A(\theta) = \int_{r=0}^{\infty}\omega(re_\theta)rdr$ the area of the intersection of Ω with a vertical half plane with polar coordinates (r,θ), $r \geqslant 0$. The polar coordinates of the horizontal projection of the centroid of that cross section are $(\rho(\theta),\theta)$ where $\rho(\theta) = \frac{1}{A(\theta)}\int_{r=0}^{\infty}\omega(re_\theta)r^2dr$

Lemma 4.2 *The value of ω at the projection centroid of a cross section of Ω is greater than $\frac{9}{25}$ times the maximum of ω. That is $\forall\theta\ \omega(\rho(\theta)e_\theta) \geqslant \frac{9}{25}\omega(0)$.*

Proof: Since $\omega(re_\theta)$ is strictly decreasing from its maximum to 0 (Theorem 3.2), there is an unique value r_θ such that $\omega(r_\theta e_\theta) = \frac{9}{25}\omega(0)$. Now we consider the fonction $\omega'_\theta(r) = \omega(0)\left(1-\frac{2r}{5r_\theta}\right)^2$ for $r \in [0, \frac{5}{2}r_\theta]$.

Using the concavity of function $\sqrt{\omega}$ (Theorem 3.1), the relative position of ω'_θ and ω are the following (see Figure 4):

$$
\begin{aligned}
\omega(0) &= \omega'_\theta(0) & &\\
\omega(re_\theta) &\geqslant \omega'_\theta(r) &,\quad & r\in[0,r_\theta]\\
\omega(re_\theta) &\leqslant \omega'_\theta(r) &,\quad & r\in[r_\theta,\frac{5}{2}r_\theta]\\
\omega(re_\theta) &= 0 &,\quad & r\in[\frac{5}{2}r_\theta,\infty]
\end{aligned}
$$

Since the weighted barycenter of function ω'_θ is

$$\int_{r=0}^{\frac{5}{2}r_\theta} \omega'_\theta(r)r^2dr \Big/ \int_{r=0}^{\frac{5}{2}r_\theta} \omega'_\theta(r)rdr = r_\theta$$

from inequalities above, we deduce that $\rho(\theta) \leqslant r_\theta$ and thus by Theorem 3.2: $\omega(\rho(\theta)e_\theta) \geqslant \omega(r_\theta) = \frac{9}{25}\omega(0)$. □

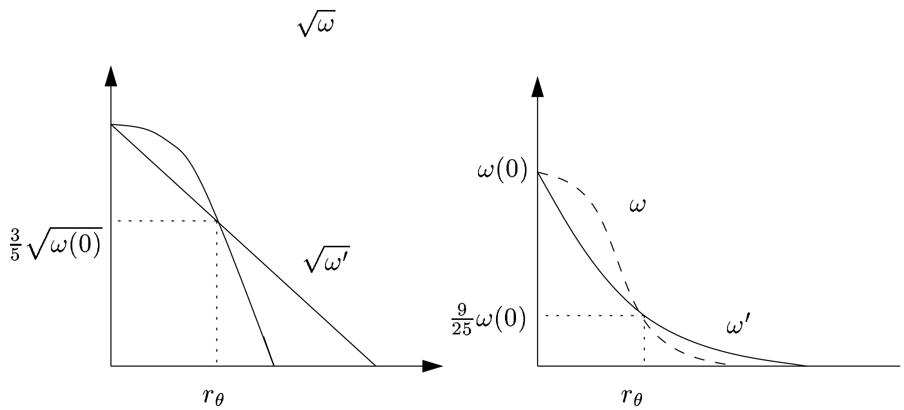

Figure 4: Relative position of ω and ω'

Lemma 4.3 *The curve $r_\theta e_\theta$ depending on θ is convex.*

Proof: It follows directly from the concavity of function $\sqrt{\omega}$ (Theorem 3.1). The curve $\theta \mapsto r_\theta e_\theta$ is the intersection of the 3D surface defined by $z = \sqrt{\omega}$ and the horizontal plane $z = \frac{3}{5}\sqrt{\omega(0)}$. □

Theorem 4.4 *The translation which matches the centroids of two convex polygons realizes an overlap area of at least $\frac{9}{25}$ the maximal overlap area.*

Proof: The overlap area when the centroids of P and Q coincide is $\omega(c_Q - c_P)$ which is $\omega(p(c_\Omega))$ by Lemma 4.1. $p(c_\Omega)$ is the centroid of points $\rho(\theta)e_\theta$ weighted by the positive function $A(\theta)$ (Equation 1). The curve $\rho(\theta)e_\theta$ is inside the convex curve $r_\theta e_\theta$ (using Lemma 4.2), which is convex by Lemma 4.3. Thus $p(c_\Omega)$ is inside the convex curve $r_\theta e_\theta$ and thus, by concavity of $\sqrt{\omega}$ (Theorem 3.1), $\omega(p(c_\Omega)) \geqslant \omega(r_\theta e_\theta) = \frac{9}{25}\omega(0)$. □

Theorem 4.5 *The translation which matches the centroids of two d-dimensional convex polyhedra realizes an overlap volume of at least* $\left(\frac{3}{d+3}\right)^d$ *times the maximal overlap volume.*

Proof: The proof is similar to the one we gave for the two dimensional case. In higher dimension we use the concavity (by Brunn-Minkowski Theorem) of function $\sqrt[d]{\omega}$ instead of $\sqrt{\omega}$. and the new definition of $\omega'(r)$ is $\omega(0)\left(1-\frac{2r}{(d+3)r_\theta}\right)^d$ which yields the claimed bound. □

4.2 Upper bounds

4.2.1 Two dimensional example

The worst known example for overlapping two polygons reaches a bound of $\frac{4}{9}$ between the maximal overlap and the overlap at the centroid position. The example is depicted in Figure 5, assuming that the small edges of triangles have length 1, and that the opposite angle is very small. The intersection at the optimal position is a small square of edge length about 1, and the intersection at the centroid position is a square of edge length about $\frac{2}{3}$.

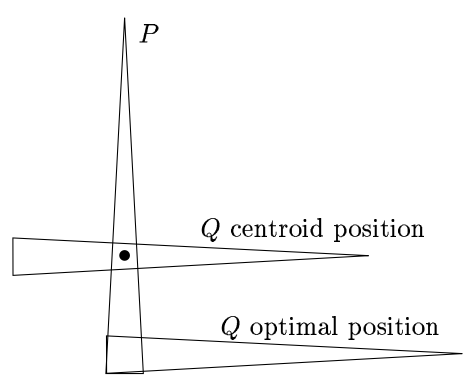

Figure 5: Example reaching $\frac{4}{9}$ upper bound.

4.2.2 Three dimensional example

Our worst example in three dimensions is drawn in Figure 6. If edge e has length 1, and the smallest face of Q as area ε then the volume of the intersection at the optimal position is ε. Let us now evaluate the volume when the centroids coincide: edge e' parallel to e in the section of P by the vertical plane through the centroid of P has length $\frac{3}{4}$, the length of the stick Q inside P is about $\frac{2}{3}$ of e' that is $\frac{1}{2}$ and finally the horizontal section of Q at its centroid is a $\frac{3}{4}$ homothet of the horizontal face of Q. Thus the volume of the intersection is $\frac{1}{2}\left(\frac{3}{4}\right)^2 \varepsilon = \frac{9}{32}\varepsilon$.

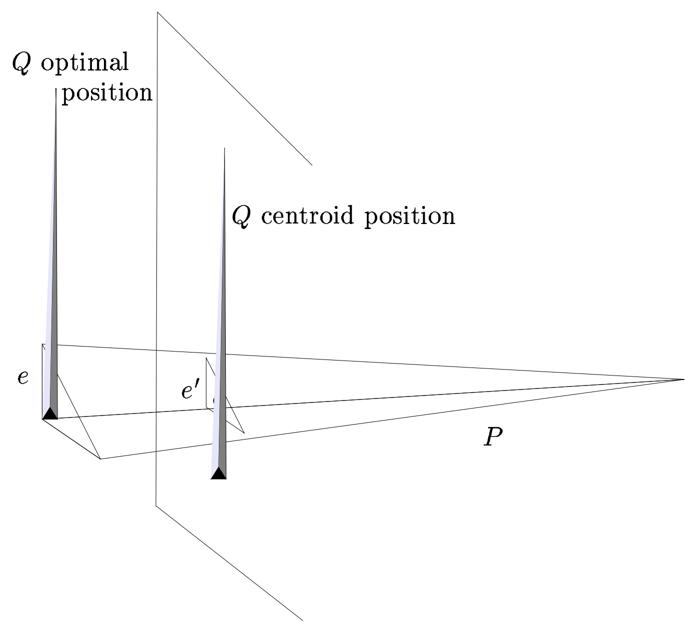

Figure 6: Example reaching $\frac{9}{32}$ upper bound.

5 Conclusions

We presented an algorithm that computes a translation of a convex polygon Q that maximizes the area of overlap with another convex polygon P. The algorithm runs in $O((n+m)\log(n+m))$ time, where n and m are the number of vertices of P and

Q, respectively. Our algorithm is based on the unimodality of the overlap function for convex polygons. We showed that particular placement for which the centroids of the polygons coincide gives an approximation of the optimal overlap area by a factor lying between $\frac{4}{9}$ and $\frac{9}{25}$. We conjecture that the lower bound $\frac{4}{9}$ is in fact tight, but the demonstration of any better bound must rely on some other facts than Theorem 3.1. In fact, it is easy to construct a tight example of bivariate function δ such that $\sqrt{\delta}$ is convex and the value of δ at its centroid is only $\frac{9}{25}$ of its maximal value, but such δ does not seem to be the representation of the ω function related to the overlap of two polygons.

The obvious next step is to develop an efficient algorithm for arbitrary simple polygons. Unfortunately, the overlap is no longer unimodal for non-convex polygons: the overlap function can have up to $\Theta(n^2m^2)$ local maxima. It seems difficult to develop an algorithm that does not inspect all local maxima. In many cases, however, the number of local maxima in the overlap function is relatively small. It would be interesting to develop an algorithm whose running time depends on the number of local maxima.

Acknowledgments

Thanks to H. Alt for helpful discussions about the centroid problem.

Acknowledgements

The authors would like to thank Machin for pointing out the applicability of [GT91] to this problem. They would also like to thank Jean-Pierre Merlet for supplying us with his interactive drawing preparation system JPdraw .

References

[AB88] F. Avnaim and J.-D. Boissonnat. Polygon placement under translation and rotation. In *Proc. 5th Sympos. Theoret. Aspects Comput. Sci.*, volume 294 of *Lecture Notes in Computer Science*, pages 322–333. Springer-Verlag, 1988.

[ABB91] H. Alt, B. Behrends, and J. Blömer. Approximate matching of polygonal shapes. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 186–193, 1991.

[ABS+96] D. Avis, P. Bose, T. Shermer, J. Snoeyink, G. Toussaint, and B. Zhu. On the sectional area of convex polytopes. In *Communication at the 12th Annu. ACM Sympos. Comput. Geom.*, page C, 1996.

[AG92] H. Alt and M. Godau. Measuring the resemblance of polygonal curves. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 102–109, 1992.

[AST92] Pankaj K. Agarwal, Micha Sharir, and Sivan Toledo. Applications of parametric searching in geometric optimization. In *Proc. 3rd ACM-SIAM Sympos. Discrete Algorithms*, pages 72–82, 1992.

[CGH+93] L. P. Chew, M. T. Goodrich, D. P. Huttenlocher, K. Kedem, J. M. Kleinberg, and D. Kravets. Geometric pattern matching under Euclidean motion. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 151–156, Waterloo, Canada, 1993.

[Cha83] B. Chazelle. The polygon containment problem. In F. P. Preparata, editor, *Computational Geometry*, volume 1 of *Advances in Computing Research*, pages 1–33. JAI Press, London, England, 1983.

[Cha92] B. Chazelle. An optimal algorithm for intersecting three-dimensional convex polyhedra. *SIAM J. Comput.*, 21(4):671–696, 1992.

[Cha93] B. Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete Comput. Geom.*, 9(2):145–158, 1993.

[CK93] L. P. Chew and K. Kedem. Placing the largest similar copy of a convex polygon among polygonal obstacles. *Comput. Geom. Theory Appl.*, 3(2):59–89, 1993.

[FJ84] G. N. Frederickson and D. B. Johnson. Generalized selection and ranking: sorted matrices. *SIAM J. Comput.*, 13:14–30, 1984.

[For85] S. J. Fortune. A fast algorithm for polygon containment by translation. In *Proc. 12th Internat. Colloq. Automata Lang. Program.*, volume 194 of *Lecture Notes in Computer Science*, pages 189–198. Springer-Verlag, 1985.

[Grü67] B. Grünbaum. *Convex Polytopes.* Wiley, New York, NY, 1967.

[GT91] M. Goodrich and R. Tamassia. Dynamic trees and dynamic point location. In *Proc. 23rd Annu. ACM Sympos. Theory Comput.*, pages 523–533, 1991.

[Hal92] D. Halperin. *Algorithmic Motion Planning via Arrangements of Curves and of Surfaces.* Ph.D. thesis, Computer Science Department, Tel-Aviv University, Tel Aviv, 1992.

[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 110–120, 1992.

[HKS93] D. P. Huttenlocher, K. Kedem, and M. Sharir. The upper envelope of Voronoi surfaces and its applications. *Discrete Comput. Geom.*, 9:267–291, 1993.

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

[MSW93] D.M. Mount, R. Silverman, and A. Wu. On the area of overlap of translated polygons. *SPIE Vision Geometry II*, 2060:254–264, 1993.

[PT92] D. Parson and C. Torras. The combinatorics of overlapping convex polygons in contact. In *Proc. 4th Canad. Conf. Comput. Geom.*, pages 83–92, 1992.

[SCK+86] M. Sharir, R. Cole, K. Kedem, D. Leven, R. Pollack, and S. Sifrony. Geometric applications of Davenport-Schinzel sequences. In *Proc. 27th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 77–86, 1986.

[Ven95] R. Venkatasubramanian. On the area of intersection of two closed 2d objects. *Information Sciences*, 82:25–44, 1995.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399