Maximum overlap area of a convex polyhedron and a convex polygon under translation

Honglin Zhu[1] **and Hyuk Jun Kweon**[2]

1 **Massachusetts Institute of Technology**
`honglinz@mit.edu`
2 **Massachusetts Institute of Technology**
`kweon@mit.edu`

Abstract

Let P be a convex polyhedron and Q be a convex polygon with n vertices in total in three-dimensional space. We present a deterministic algorithm that finds a translation vector $v \in \mathbb{R}^3$ maximizing the overlap area $|P \cap (Q + v)|$ in $O(n \log^2 n)$ time. We then apply our algorithm to solve two related problems. We give an $O(n \log^3 n)$ time algorithm that finds the maximum overlap area of three convex polygons with n vertices in total. We also give an $O(n \log^2 n)$ time algorithm that minimizes the symmetric difference of two convex polygons under scaling and translation.

Related Version A full version of the paper is available at `https://arxiv.org/abs/2301.02949`.

1 Introduction

Shape matching is an important topic in computational geometry, with useful applications in areas such as computer graphics. In a typical problem of shape matching, we are supplied two or more shapes, and we want to determine how much the shapes resemble each other. More precisely, given a similarity measure and a set of allowed transformations, we want to transform the shapes to maximize their similarity measure.

While there are many candidates for the similarity measure, we focus on the area/volume of overlap or of symmetric difference. The advantage to these is that they are robust against noise on the boundary of the images [6]. For previous results in this area, see [6, 1, 3, 4, 2, 12].

While many have studied the matching problem for two convex polytopes of the same dimension, few have examined the problem for polytopes of different dimensions or matching more than two polytopes. In this paper, we present a deterministic algorithm for finding the maximum overlap area of a convex polyhedron and a convex polygon under translation in three-dimensional space. Using this algorithm, we also solve the problems of maximizing the overlap of three convex polygons and of minimizing the symmetric difference of two convex polygons under homothety.

2 Preliminaries

Let $P \subset \mathbb{R}^3$ be a convex polyhedron and $Q \subset \mathbb{R}^2$ be a convex polygon with n vertices in total. Throughout the paper, we assume that Q is in the xy-plane, and that the lowest point of P is on the xy-plane. We want to find a translation vector $v = (x, y, z) \in \mathbb{R}^3$ that maximizes the overlap area $f(v) = |P \cap (Q + v)|$.

It is easy to observe that $f(v)$ is continuous and piecewise quadratic on the interior of its support. As noted in [6, 1, 3], f is smooth on a region R if $P \cap (Q + v)$ is combinatorially equivalent for all $v \in R$, that is, if we have the same set of face-edge incidences between P and Q. Following the convention of [1], we call the polygons that form the boundaries of these regions the *event polygons*, and as in [6], we call the space of translations of Q the

39th European Workshop on Computational Geometry, Barcelona, Spain, March 29–31, 2023.
This is an extended abstract of a presentation given at EuroCG'23. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

configuration space. The arrangement of the event polygons partition the configuration space into cells with disjoint interiors. The overlap function $f(v)$ is quadratic on each cell. Thus, to locate a translation maximizing f, we need to characterize the event polygons.

For two sets $A, B \subset \mathbb{R}^d$, we write the *Minkowski sum* of A and B as $A + B := \{a + b | a \in A, b \in B\}$. We also write $A - B$ for the Minkowski sum of A with $-B = \{-b | b \in B\}$. We categorize the event polygons into three types and describe them in terms of Minkowski sums:

(I) When $Q + v$ contains a vertex of P. For each vertex u of P, we have an event polygon $u - Q$. There are $O(n)$ event polygons of this type.

(II) When a vertex of $Q + v$ is contained in a face of P. For each face F of P and each vertex v of Q, we have an event polygon $F - v$. There are $O(n^2)$ event polygons of this type.

(III) When an edge of $Q + v$ intersects an edge of P. For each edge e of P and each edge e' of Q, we have an event polygon $e - e'$. There are $O(n^2)$ event polygons of this type.

The reason that convexity is fundamental is due to the following standard fact, as noted and proved in [6, 12].

▶ **Proposition 2.1.** *Let P be a d'-dimensional convex polytope and let Q be a d-dimensional convex polytope. Suppose $d' \geq d$. Let $f(v) = \mathrm{Vol}(P \cap (Q + v))$ be the volume of the overlap function. Then, $f(v)^{1/d}$ is concave on its support* $\mathrm{supp}(f) = \{v | f(v) > 0\}$.

As in [5], we say a function $f : \mathbb{R} \to \mathbb{R}$ is *unimodal* (resp. *strictly unimodal*) if it increases (resp. strictly increases) to a maximum value, possibly stays there for some interval, and then decreases (resp. strictly decreases). Furthermore, we say a function $f : \mathbb{R}^d \to \mathbb{R}$ is unimodal (resp. strictly unimodal) if its restriction to any line is unimodal (resp. strictly unimodal).

▶ **Corollary 2.2** ([5])**.** *For any line l parameterized by $l = p + vt$ in $\mathbb{R}^{d'}$, the function $f_l(t) = f(p + vt)$ is strictly unimodal.*

We introduce a divide-and-conquer technique that we apply in our algorithm.

▶ **Lemma 2.3** ([8])**.** *Given n hyperplanes in $\mathbb{R}^d$ and a region $R \subset \mathbb{R}^d$, a $(1/r)$-cutting is a collection of simplices with disjoint interiors, which together cover R and such that the interior of each simplex intersects at most n/r hyperplanes. A $(1/r)$-cutting of size $O(r^d)$ can be computed deterministically in $O(nr^{d-1})$ time. In addition, the set of hyperplanes intersecting each simplex of the cutting is reported in the same time.*

Another technique that we use in our algorithm is a generalization of Megiddo's prune-and-search [11]. This technique is of independent interest and can likely be applied to other problems.

▶ **Theorem 2.4.** *Let $S = \bigcup_{i=1}^{n} S_i$ be a union of n sets of $O(m)$ parallel lines in the plane, none of which are parallel to the x-axis, and suppose the lines in each S_i are indexed from left to right.*

Suppose there is an unknown point $p^* \in \mathbb{R}^2$ and we are given an oracle that decides in time T the relative position of p^* to any line in the plane. Then we can find the relative position of p^* to every line in S in $O(n \log^2 m + (T + n) \log(mn))$ time.

3 Maximum overlap of convex polyhedron and convex polygon

In this section, we prove our main result:

▶ **Theorem 3.1.** *Let P be a convex polyhedron and Q a convex polygon with n vertices in total. We can find a vector $v \in \mathbb{R}^3$ that maximizes the overlap area $|P \cap (Q+v)|$ in $O(n \log^2 n)$ time.*

Following the convention in [6], we call a translation that maximizes the overlap function f a *goal placement*. In the algorithm, we keep track of a closed *target region* R which we know contains a goal placement and decrease its size until for each event polygon F, either $F \cap \text{interior}(R) = \varnothing$ or $F \supset R$. Then, f is quadratic on R and we can find the maximum of f on R using standard calculus. Thus, the goal of our algorithm is to efficiently trim R to eliminate event polygons that intersect it.

In the beginning of the algorithm, the target region is the interior of the Minkowski sum $P - Q$, where the overlap function is positive. By the unimodality of the overlap function, the set of goal placements is convex. Thus, for a plane in the configuration space, either it contains a goal placement, or all goal placements lie on one of the two open half spaces separated by the plane. If we have a way of knowing which case it is for any plane, we can decrease the size of our target region by cutting it with planes and finding the piece to recurse. More precisely, we need a subroutine **PlaneDecision** that decides the relative position of the set of goal placements to a plane S.

Whenever **PlaneDecision** reports that a goal placement is found on a plane, we can let the algorithm terminate. Thus, we can assume it always reports a half-space containing a goal placement.

As in Algorithm 1, we break down our algorithm into three stages.

Algorithm 1: Pseudocode for Theorem 3.1

input : A convex polyhedron $P \in \mathbb{R}^3$ and a convex polygon $Q \in \mathbb{R}^3$ with n vertices in total
output : A translation $v \in \mathbb{R}^3$ maximizing the area $|P \cap (Q+v)|$

1 Locate a horizontal slice containing a goal placement that does not contain any vertices of P and replace P by this slice of P
2 Find a "tube" $D + l_y$ whose interior contains a goal placement and intersects $O(n)$ event polygons, where D is a triangle in the xz-plane and l_y is the y-axis
3 Recursively construct a $(1/2)$-cutting of the target region $D + l_y$ to find a simplex containing a goal placement that does not intersect any event polygon

3.1 Stage 1

In the first stage of our algorithm, we make use of [6] to simplify our problem so that P can be taken as a convex polyhedron with all of its vertices on two horizontal planes.

We sort the vertices of P by z-coordinate in increasing order and sort the vertices of Q in counterclockwise order. Next, we trim the target region with horizontal planes (planes parallel to the xy-plane) to get to a slice that does not contain any vertices of P.

▶ **Lemma 3.2.** *In $O(n \log^2 n)$ time, we can locate a strip $R = \{(x, y, z) | z \in [z_0, z_1]\}$ whose interior contains a goal placement and P has no vertices with $z \in [z_0, z_1]$.*

By Chazelle's algorithm [7], the convex polyhedron $P' = \{(x, y, z) \in P | z \in [z_0, z_1]\}$ can be computed in $O(n)$ time. From now on, we replace P with P' (see Figure 1). Without loss of generality, assume $z_0 = 0$ and $z_1 = 1$.

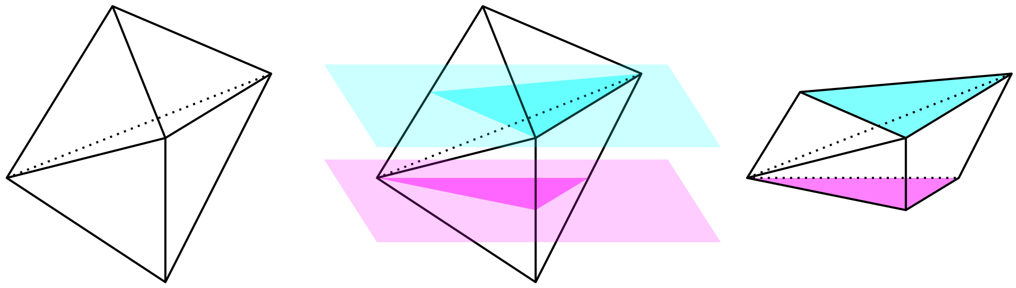


Figure 1 The slice of P with $z \in [z_0, z_1]$.

The region in the configuration space where $|P \cap (Q+v)| > 0$ is the Minkowski sum $P - Q$. Since P only has two levels $P_0 = \{(x,y,z) \in P | z = 0\}$ and $P_1 = \{(x,y,z) \in P | z = 1\}$ that contain vertices, the Minkowski sum $P - Q$ is simply the convex hull of $(P_0 - Q) \cup (P_1 - Q)$, which has $O(n)$ vertices. We can compute $P_0 - Q$ and $P_1 - Q$ in $O(n)$ time and compute their convex hull in $O(n \log n)$ time by Chazelle's algorithm [9].

3.2 PlaneDecision

Due to space constraints, we will not present the algorithm **PlaneDecision**. For details, see the full version of the paper [10].

We present a perturbation method to reduce the problem of deciding the relative position of the goal placement to an arbitrary plane to finding the maximum of the overlap over an arbitrary plane.

▶ **Lemma 3.3.** *Suppose we can compute $\max_{v \in S} f(v)$ for any plane $S \subset \mathbb{R}^3$ in time T, then we can perform **PlaneDecision** for any plane in time $O(T)$.*

Proof. The idea is to compute $\max_{v \in S'} f(v)$ for certain S' that are perturbed slightly from S to see in which direction relative to S does f increase.

We compute over an extension of the reals $\mathbb{R}[\omega]/(\omega^3)$, where $\omega > 0$ is smaller than any real number. Let $A > 0$ be the maximum of f over a plane S. Let S_+ and S_- be the two planes parallel to S that have distance ω from S. We compute $A_+ = \max_{v \in S_+} f(v)$ and $A_- = \max_{v \in S_-} f(v)$ in $O(T)$ time. Since f is piecewise quadratic, A_+ and A_- as symbolic expression will only involve quadratic terms in ω. Since f is strictly unimodal on $P - Q$, there are three possibilities:

1. If $A_+ > A$, then halfspace on the side of S_+ contains the set of goal placements.
2. If $A_- > A$, then halfspace on the side of S_- contains the set of goal placements.
3. If $A \geq A_+$ and $A \geq A_-$, then A is the global maximum of f.

Thus, in $O(T)$ time, we can finish **PlaneDecision**. ◀

With Theorem 3.3, it suffices for us to give an algorithm finding the maximum of the overlap over any plane. Just like in our main algorithm, we want to prune the configuration space (now restricted to a plane), until we locate a region that contains the maximum and that does not intersect any event polygon. In Algorithm 2, we give an outline for **PlaneDecision**:

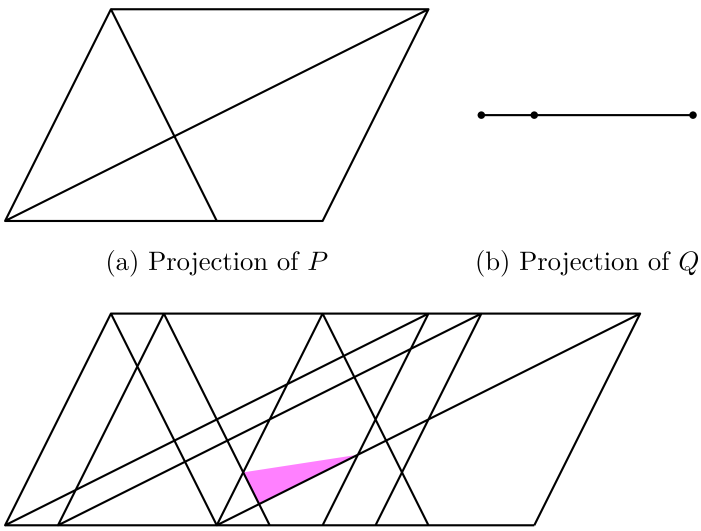
▶ **Proposition 3.4.** *For a plane S, we can perform **PlaneDecision** in $O(n \log n)$ time.*

Algorithm 2: Pseudocode for **PlaneDecision**

input : A plane $S \subset \mathbb{R}^3$
output : A translation $v \in S$ maximizing the area $|P \cap (Q+v)|$
1 Compute $S \cap (P-Q)$ and set it to be our initial target region.
2 Locate a strip on S containing a good placement whose interior intersects $O(n)$ event polygons.
3 Recursively construct a (1/2)-cutting of the strip to find a triangle containing a good placement that does not intersect any event polygon

3.3 Stage 2

With the general **PlaneDecision** at our disposal, we now move on to Stage 2, the main component of our algorithm. We project the entire configuration space and the event polygons onto the xz-plane in order to find a target region D whose preimage $D + l_y$ intersects few event polygons, where l_y is the y-axis (see Figure 2).



(a) Projection of P (b) Projection of Q

(c) Projection of the configuration space, and the target region D

Figure 2 Projecting onto the xz-plane.

The non-horizontal edges of the event polygons project to segments on the strip $0 < z < 1$ on the xz-plane. We characterize our desired region D in the following lemma.

▶ **Lemma 3.5.** *For a region D that does not intersect any of the segments that are the projections of the non-horizontal edges of the event polygons, the preimage $D + l_y$ intersects $O(n)$ event polygons.*

Now it remains to efficiently find such a region D with $D + l_y$ containing a goal placement and compute the $O(n)$ event polygons that intersect its interior.

▶ **Lemma 3.6.** *In $O(n \log^2 n)$ time, we can find a triangle D in the xz-plane such that the interior of $D + l_y$ contains a goal placement and intersects $O(n)$ event polygons. We can compute these $O(n)$ event polygons in the same time bound.*

3.4 Stage 3

Now, we have a target region $R = D + l_y$ whose interior contains a goal placement, and we have the $O(n)$ event polygons that intersect it.

▶ **Lemma 3.7.** *In $O(n \log^2 n)$ time, we can find a region $R' \subset R$ containing a goal placement that does not intersect any of the $O(n)$ event polygons.*

Finally, since the overlap function is quadratic on our final region R', we can solve for the maximum using standard calculus. This concludes the proof of Theorem 3.1.

4 Applications

We present two applications of Theorem 3.1 to other problems in computational geometry. First, we give a deterministic algorithm for maximizing the overlap of three convex polygons.

▶ **Theorem 4.1.** *Let P, Q, R be three convex polygons with n vertices in total in the plane. We can find a pair of translations $(v_Q, v_R) \in \mathbb{R}^4$ that maximizes the overlap area $|P \cap (Q + v_Q) \cap (R + v_R)|$ in $O(n \log^3 n)$ time.*

In this problem, the configuration space is four-dimensional. An easy extension of Proposition 2.1 and Theorem 2.2 shows that the function of overlap area is again unimodal. This time, we have four-dimensional *event polyhedra* instead of event polygons that divide the configuration space into four-dimensional cells on which $g(v_Q, v_R)$ is quadratic. We call a hyperplane containing an event polyhedron an *event hyperplane*, and they are defined by two types of events:

(I) When one vertex of P, $Q + v_Q$ or $R + v_R$ lies on an edge of another polygon. There are $O(n)$ groups of $O(n)$ parallel event hyperplanes of this type.
(II) When an edge from each of the three polygons intersect at one point. There are $O(n^3)$ event hyperplanes of this type.

To overcome the difficulty of dealing with the $O(n^3)$ event hyperplanes of type (II), we first prune the configuration space to a region intersecting no event hyperplanes of type (I). We then show that the resulting region only intersects $O(n)$ event hyperplanes of type (II), at which point we can use Theorem 2.3 iteratively to finish.

Observe that maximizing the overlap over a hyperplane in which two polygons move relatively in a line and the other polygon moves freely corresponds precisely to the problem of overlapping a convex polyhedron (whose cross-sections are the intersections of the two polygons moving in a line) and a convex polygon (the third polygon). Thus, Theorem 3.1 gives us a kind of "**HyperplaneDecision**" which we can use to prune the event hyperplanes of type (I).

We also give a deterministic $O(n \log^2 n)$ time algorithm for minimizing the symmetric difference of two convex polygons under homothety (a scaling and a translation), which is an improvement to Yon et al.'s $O(n \log^3 n)$ time algorithm [12].

▶ **Theorem 4.2.** *Let P and Q be convex polygons with n vertices in total. Then we can find a homothety φ that minimizes the area of symmetric difference $|P \setminus \varphi(Q)| + |\varphi(Q) \setminus P|$ in $O(n \log^2 n)$ time.*

We want to minimize the function $h(\varphi) = h(x, y, \lambda) = |P \setminus \varphi(Q)| + |\varphi(Q) \setminus P|$, where $\varphi(Q) = \lambda Q + (x, y)$. We can rewrite this as $h(\varphi) = |P| + |Q|\lambda^2 - 2|P \cap \varphi(Q)|$. Thus, minimizing h is the same as maximizing the function $f(\varphi) = |P \cap \varphi(Q)| - c\lambda^2$, where $c = \frac{1}{2}|Q|$.

Consider the cone $C = \{(x, y, \lambda) | \lambda \in [0, M], (x, y) \in \lambda Q\}$, where $M = \sqrt{|P|/c}$ (see Figure 3). Then f is negative for $\lambda > M$ so it is never maximized. We also put P into $\mathbb{R}^3$ by $P = \{(x, y, 0)|(x, y) \in P\}$. Since $f(x, y, \lambda) = |C \cap (P + (-x, -y, \lambda))| - c\lambda^2$, the problem reduces to maximizing the overlap area of the cone C and P under translation subtracted by a quadratic function. Some modification of Theorem 3.1 gives Theorem 4.2.

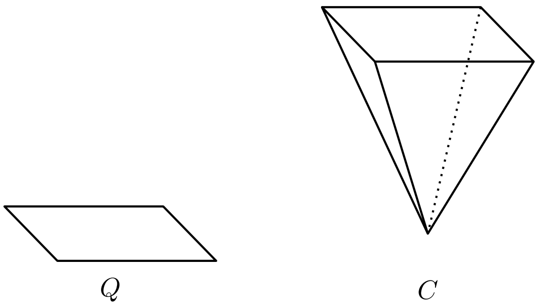


Figure 3 Formation of the cone C.

References

1 Hee-Kap Ahn, Peter Brass, and Chan-Su Shin. Maximum overlap and minimum convex hull of two convex polyhedra under translations. *Comput. Geom.*, 40(2):171–177, 2008. doi:10.1016/j.comgeo.2007.08.001.

2 Hee-Kap Ahn, Siu-Wing Cheng, Hyuk Jun Kweon, and Juyoung Yon. Overlap of convex polytopes under rigid motion. *Comput. Geom.*, 47(1):15–24, 2014. doi:10.1016/j.comgeo.2013.08.001.

3 Hee-Kap Ahn, Siu-Wing Cheng, and Iris Reinbacher. Maximum overlap of convex polytopes under translation. *Comput. Geom.*, 46(5):552–565, 2013. doi:10.1016/j.comgeo.2011.11.003.

4 Hee-Kap Ahn, Otfried Cheong, Chong-Dae Park, Chan-Su Shin, and Antoine Vigneron. Maximizing the overlap of two planar convex sets under rigid motions. *Comput. Geom.*, 37(1):3–15, 2007. doi:10.1016/j.comgeo.2006.01.005.

5 David Avis, Prosenjit Bose, Thomas C. Shermer, Jack Snoeyink, Godfried Toussaint, and Binhai Zhu. On the sectional area of convex polytopes. In *Communication at the 12th Annu. ACM Sympos. Comput. Geom.*, page C. Association for Computing Machinery, New York, NY, 1996.

6 Mark de Berg, Olivier Devillers, Marc van Kreveld, Otfried Schwarzkopf, and Monique Teillaud. Computing the maximum overlap of two convex polygons under translations. In *International Symposium on Algorithms and Computation*, pages 126–135. Springer, 1996.

7 Bernard Chazelle. An optimal algorithm for intersecting three-dimensional convex polyhedra. *SIAM J. Comput.*, 21(4):671–696, 1992. doi:10.1137/0221041.

8 Bernard Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete Comput. Geom.*, 9(2):145–158, 1993. doi:10.1007/BF02189314.

9 Bernard Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete Comput. Geom.*, 10(4):377–409, 1993. doi:10.1007/BF02573985.

10 Hyuk Jun Kweon and Honglin Zhu. Maximum overlap area of a convex polyhedron and a convex polygon under translation, 2023. URL: https://arxiv.org/abs/2301.02949, doi:10.48550/ARXIV.2301.02949.

11 Nimrod Megiddo. Linear programming in linear time when the dimension is fixed. *J. Assoc. Comput. Mach.*, 31(1):114–127, 1984. doi:10.1145/2422.322418.

12 Juyoung Yon, Sang Won Bae, Siu-Wing Cheng, Otfried Cheong, and Bryan T. Wilkinson. Approximating convex shapes with respect to symmetric difference under homotheties. In *32nd International Symposium on Computational Geometry*, volume 51 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 63, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2016.