Maximum Overlap of Convex Polytopes under Translation*

Hee-Kap Ahn[†] Siu-Wing Cheng[‡] Iris Reinbacher[†]

Abstract

We study the problem of maximizing the overlap of two convex polytopes under translation in $\mathbb{R}^d$ for some constant $d \geq 3$. Let n be the number of bounding hyperplanes of the polytopes. We present an algorithm that, for any $\varepsilon > 0$, finds an overlap at least the optimum minus ε and reports the translation realizing it. The running time is $O(n^{\lfloor d/2 \rfloor + 1} \log^d n)$ with probability at least $1 - n^{-O(1)}$, which can be improved to $O(n \log^{3.5} n)$ in $\mathbb{R}^3$. The time complexity analysis depends on a bounded incidence condition that we enforce with probability one by randomly perturbing the input polytopes. The perturbation causes an additive error ε, which can be made arbitrarily small by decreasing the perturbation magnitude. Our algorithm in fact computes the maximum overlap of the perturbed polytopes. The running time bounds, the probability bound, and the big-O constants in these bounds are independent of ε.

1 Introduction

Many applications perform geometric shape matching to find a transformation of one shape in order to maximize some similarity measure with another shape. The problem of matching convex shapes has been used in tracking regions in an image sequence [15] and measuring symmetry of a convex body [12]. One robust similarity measure for two shapes is their *overlap*—the volume of their intersection [18]. In this paper, we consider maximizing the overlap of two convex polytopes under translation in $\mathbb{R}^d$ for $d \geq 3$. The dimension d is treated as a constant and so is any value depending on d alone.

In the plane, the maximum overlap problem has been studied for convex and simple polygons. Let n be the number of input polygon edges. De Berg et al. [3] can maximize the overlap of two convex polygons under translation in $O(n \log n)$ time. Mount et al. [16] can do the same for two simple polygons in $O(n^4)$ time. When both rotation and translation are allowed, Ahn et al. [2] can align two convex polygons with an overlap at least $1 - \varepsilon$ times the optimum for any $\varepsilon \in (0, 1)$. The running time of their algorithm is $O((1/\varepsilon) \log n + (1/\varepsilon^2) \log(1/\varepsilon))$, assuming that there are two input arrays, each storing the polygon vertices in order around the boundary. If only translation is allowed, Ahn et al. can improve the running time to $O((1/\varepsilon) \log n + (1/\varepsilon) \log(1/\varepsilon))$.

The maximum overlap problem for convex polytopes under translation in $\mathbb{R}^d$ for $d \geq 3$ has been studied by Ahn et al. [1] and Fukuda and Uno [10]. Let n be the number of hyperplanes defining the convex polytopes. Ahn et al.'s algorithm finds the maximum overlap of two convex polytopes under translation in $O(n^{(d^2+d-3)/2} \log^{d+1} n)$ expected time. Given k convex polytopes

*Research of Ahn was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2007-331-D00372). Research of Cheng was partly supported by Research Grant Council, Hong Kong, China (project no. 612107). Research of Reinbacher was partly supported by a postdoc matching fund of HKUST.

[†]Department of Computer Science and Engineering, POSTECH, Korea. Email: `heekap@postech.ac.kr`, `irisrein@postech.ac.kr`

[‡]Department of Computer Science and Engineering, HKUST, Hong Kong. Email: `scheng@cse.ust.hk`

for some constant $k \geq 2$, Fukuda and Uno can translate them to give an overlap at least $\texttt{opt} - \varepsilon$ for any $\varepsilon > 0$, where $\texttt{opt}$ denotes the maximum overlap. They require $O(\log(\texttt{opt}/\varepsilon))$ calls to a subroutine that returns the value and the gradient of the overlap function for given translations of the polytopes. Some critical details of this subroutine are missing though. In any case, the running time does not depend on the combinatorial input size n alone. Fukuda and Uno also gave an algorithm to find the maximum overlap of k possibly non-convex polytopes under translation in $O(n^{kd^2+d})$ time.

Vigneron [19] studied the optimization of algebraic functions and one of the applications is the alignment of two possibly non-convex polytopes under rigid motion. For any $\varepsilon \in (0,1)$ and for any two convex polytopes with n defining hyperplanes, Vigneron's method can return in $O\big(\varepsilon^{-\Theta(d^2)} n^{\Theta(d^3)} (\log \frac{n}{\varepsilon})^{\Theta(d^2)}\big)$ time an overlap under rigid motion that is at least $1-\varepsilon$ times the optimum.

We give a new algorithm for the maximum overlap problem for two convex polytopes under translation in $\mathbb{R}^d$ for $d \geq 3$. Our model of computation is the real-RAM model in which the operations $(+, -, \times, /)$ can be performed in constant time. We also make the standard assumption that it takes $O(1)$ time to solve a system of $O(1)$ polynomials of fixed degree in $O(1)$ variables. For any $\varepsilon > 0$, we can find an overlap at least the optimum minus ε and report the translation realizing it. Our algorithm runs in $O(n^{\lfloor d/2 \rfloor + 1} \log^d n)$ time with probability $1 - n^{-O(1)}$, which can be improved to $O(n \log^{3.5} n)$ in $\mathbb{R}^3$. The time complexity analysis depends on a bounded incidence condition, which may fail in degenerate situations. We enforce it with probability one by randomly perturbing the input polytopes. This causes an additive error ε, which can be made arbitrarily small by decreasing the perturbation magnitude. Our algorithm in fact computes the maximum overlap of the perturbed polytopes. The running time bounds, the probability bound, and the big-O constants in these bounds are independent of ε.

2 Preliminaries

Let X be a subset of a topological space. We use $\mathrm{bd}(X)$ to denote the *boundary* of X. Notice that $\mathrm{bd}(X)$ is empty if X is a point or an open set. The *interior* of X, denoted by $\mathrm{int}(X)$, is equal to $X \setminus \mathrm{bd}(X)$. The *closure* of X, denoted by $\mathrm{cl}(X)$, is the smallest closed set containing X. The *Minkowski sum* of two subsets X and Y of $\mathbb{R}^d$ is defined as $X \oplus Y = \{x+y : x \in X, y \in Y\}$. So $\dim(X \oplus Y) \leq \dim(X) + \dim(Y)$. For any $\alpha \in \mathbb{R}^d$, we have $X \oplus \{\alpha\} = X + \alpha$.

An *i-flat* is $L + v$ for some i-dimensional linear subspace L and for some point $v \in \mathbb{R}^d$, i.e., a copy of L translated by the vector v. A *hyperplane* in $\mathbb{R}^d$ is a $(d-1)$-flat. Given a subset $X \subset \mathbb{R}^d$, its *affine hull* $\mathrm{aff}(X)$ is the flat of the lowest dimension containing X. For example, if X is a line segment, then $\mathrm{aff}(X)$ is its supporting line; if X is a polygon, then $\mathrm{aff}(X)$ is its supporting plane.

A convex polytope P in $\mathbb{R}^d$ is the common intersection of (closed) halfspaces. These are the *bounding halfspaces* and their boundaries are the *bounding hyperplanes* of P. Assume that P has dimension d. For $k \in [0,d]$, a *k-face* of P is the k-dimensional common intersection of P and some bounding hyperplane(s). Taking no bounding hyperplane in the intersection gives the d-face, which is P itself. We follow the convention to call the 0-faces *vertices*, the 1-faces *edges*, and the $(d-1)$-faces *facets*. We use $\mathrm{faces}(P)$ to denote the set of k-faces of P for $k \in [0,d]$. The faces with dimensions less than d are called *proper faces* and they are subsets of $\mathrm{bd}(P)$. In non-degenerate situations, a k-face lies in exactly $d-k$ bounding hyperplanes. In degenerate situations, a k-face may lie in more than $d-k$ bounding hyperplanes. Each proper face of P is a convex polytope of dimension less than d.

An *i-simplex* is an i-dimensional convex polytope with exactly $i+1$ vertices.

Let $\mathcal{F}$ be a finite family of convex subsets of $\mathbb{R}^d$, each of dimension $d-1$ or less. The *arrangement* $\mathrm{Arr}(\mathcal{F})$ of $\mathcal{F}$ is a partition of $\mathbb{R}^d$ into disjoint *cells*. A cell is either a connected component in $\mathbb{R}^d \setminus (\bigcup_{S\in\mathcal{F}} S)$, or a maximal collection of points in $\bigcup_{S\in\mathcal{F}} S$ that belong to the same elements of $\mathcal{F}$. The dimensions of the cells can range from 0 to d.

Lemmas 1 and 2 below state the results on the ε-net theory [11, 13] and cuttings [6] that we use heavily. Let H be a set of hyperplanes. For any $r \in (0, |H|)$, a simplicial complex in $\mathbb{R}^d$ is called a $(1/r)$-*cutting* of H if at most $|H|/r$ hyperplanes in H intersect $\mathrm{int}(\tau)$ for any d-simplex τ in the simplicial complex.

Lemma 1 *Let H be a multiset of hyperplanes in $\mathbb{R}^d$. Let $r \in (0, |H|)$ and $\delta \in (0, 1)$ be two parameters. There is a number $j_{d,r,\delta} = \Theta(dr \log(dr/\delta))$ such that, if we draw $j_{d,r,\delta}$ hyperplanes from H uniformly at random and form an arrangement A of the hyperplanes drawn (after removing duplicates), then it holds with probability at least $1 - \delta$ that at most $|H|/r$ hyperplanes in H intersect* $\mathrm{int}(\tau)$ *for any d-simplex τ whose interior lies in a cell of A.*

Remark. In the lemma above, if we want a probability bound of $1 - |H|^{-O(1)}$, we need to draw $O(r \log |H|)$ hyperplanes to guarantee that at most $|H|/r$ hyperplanes intersect $\mathrm{int}(\tau)$.

Lemma 2 *Let H be a set of hyperplanes in $\mathbb{R}^d$. For any $r \in (0, |H|)$, a $(1/r)$-cutting of H of size $O(r^d)$ can be constructed in $O(|H|r^{d-1})$ time. Within the same time bound, one can store at each d-simplex in the $(1/r)$-cutting the hyperplanes in H that intersect its interior.*

3 Overview

Let P_1 and P_2 denote the two input convex polytopes. They are specified by n distinct bounding hyperplanes. The complexity of P_j, $j \in \{1, 2\}$, is the number of its faces, which is $O(n^{\lfloor d/2 \rfloor})$ [9]. We always translate P_1 and keep P_2 stationary. We need the following definitions.

- For any vector $\alpha \in \mathbb{R}^d$, Q_α denotes the common intersection $(P_1 + \alpha) \cap P_2$.
- For any $f \in \mathrm{faces}(P_1)$ and $g \in \mathrm{faces}(P_2)$, $\gamma_{f,g}$ denotes the set $\{\alpha \in \mathbb{R}^d : (\mathrm{int}(f) + \alpha) \cap \mathrm{int}(g) \neq \emptyset\}$, which is a single point or an open convex set.
- Γ denotes the set $\{\, \gamma_{f,g} : \dim(\gamma_{f,g}) < d \,\}$.

Observe that $\alpha \in \gamma_{f,g}$ if and only if there exists $x \in \mathrm{int}(f)$ such that $x + \alpha \in \mathrm{int}(g)$, which is equivalent to $\alpha = (-x) + y$ for some $y \in \mathrm{int}(g)$. In other words, $\gamma_{f,g} = (-\mathrm{int}(f)) \oplus \mathrm{int}(g)$. Figure 1 gives some illustrations of $\gamma_{f,g}$.

The dimension of $\gamma_{f,g}$ is less than d if for any $\alpha \in \mathbb{R}^d$ such that $(\mathrm{int}(f) + \alpha) \cap \mathrm{int}(g) \neq \emptyset$, we can perturb α slightly to α' such that $(\mathrm{int}(f) + \alpha') \cap \mathrm{int}(g) = \emptyset$. Thus, if we move a point α in $\mathbb{R}^d$, there is a combinatorial change in Q_α whenever the point α crosses an element of Γ. There is no combinatorial change in Q_α if the point α varies within a cell in $\mathrm{Arr}(\Gamma)$. Let $\mathrm{vol}(Q_\alpha)$ denote the volume of Q_α. The function $\mathrm{vol}(Q_\alpha)^{1/d}$ is concave over $\{\, \alpha \in \mathbb{R}^d : Q_\alpha \neq \emptyset \,\}$ [17].

We follow the high level approach in the algorithm of de Berg et al. for convex polygons [3]. We refer to their algorithm as Polygon. One can extend Polygon directly to higher dimensions, but this gives an $\Omega(n^{2\lfloor d/2 \rfloor})$ running time in the worst case as we explain below. For $d = 2$, Γ consists of open line segments (translations that place a vertex of P_1 in the interior of an edge of P_2 and vice versa) and the endpoints of the closure of these line segments (translations that align vertices of P_1 and P_2). Let L be the set of horizontal lines through the segment endpoints in Γ. Each line in L is the set of translations that place a vertex of P_1 at the same height of some vertex of P_2. The arrangement of Γ is divided into strips by the lines in L. Polygon locates the strip containing the solution by probing L in a binary search manner. In each probe, Polygon solves the maximum overlap problem for P_1 and P_2 with translations restricted to a line $\ell \in L$, and decide whether the solution for the original 2D problem lies above

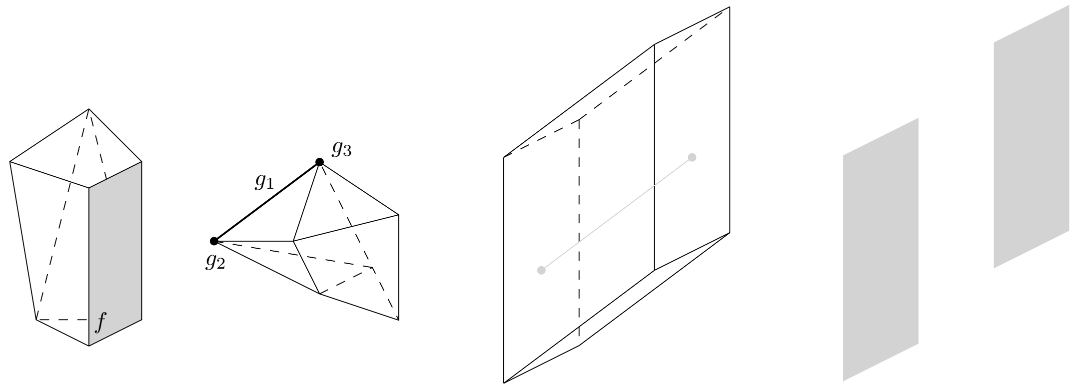

Figure 1: Illustrations of $\gamma_{f,g}$ and Γ in $\mathbb{R}^3$. Left: f is a facet of P_1 and g_1 is an edge of P_2 with endpoints g_2 and g_3. Middle: The interior of the prism is γ_{f,g_1}. The interior of the left and right vertical facets of the prism are γ_{f,g_2} and γ_{f,g_3}, respectively. Right: Γ contains γ_{f,g_2} and γ_{f,g_3} but not γ_{f,g_1} because $\dim(\gamma_{f,g_1}) = 3$.

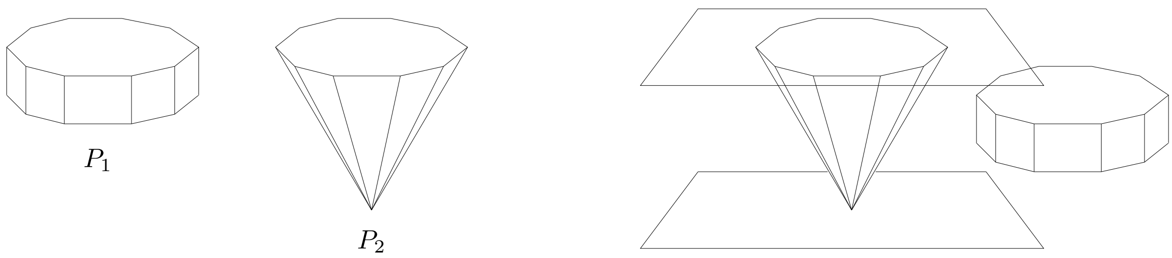

Figure 2: Left: P_1 and P_2. Right: The translations in the slab S place P_1 between the horizontal planes through the top facet and bottom vertex of P_2.

or below ℓ. Let S be the strip obtained by the binary search. De Berg et al. showed that S is stabbed by $O(n)$ open line segments in Γ. Polygon scans the vertices and edges of P_1 and P_2 in order from top to bottom to find the $O(n)$ pairs of vertices and edges that induce the open line segments in Γ stabbing S. Then, exploiting the concavity of $\mathrm{vol}(Q_\alpha)^{1/2}$, Polygon constructs a sequence of cuttings (see our Lemma 2) to prune the search space to the cell in $\mathrm{Arr}(\Gamma) \cap S$ that contains the solution for the 2D maximum overlap problem.

For $d \geq 3$, the lines in L become parallel hyperplanes and each hyperplane is the set of translations that place a vertex of P_1 at the same height as some vertex of P_2. The hyperplanes in L cut $\mathrm{Arr}(\Gamma)$ into d-dimensional slabs. One can still locate the slab S containing the solution for the maximum overlap problem by a binary search. However, for a vertex v of P_1, the translated slab $v + S$ can cross $\Theta(n^{\lfloor d/2 \rfloor})$ faces of P_2, so v induces $\Theta(n^{\lfloor d/2 \rfloor})$ elements of Γ that stab S. Summing over all faces of P_1, there can be $\Theta\big(n^{2\lfloor d/2 \rfloor}\big)$ elements of Γ that stab S. Hence, it would take $\Omega\big(n^{2\lfloor d/2 \rfloor}\big)$ time to construct a cutting on the elements of Γ stabbing S. Figure 2 shows such a bad case in $\mathbb{R}^3$. In the figure, the top and bottom facets of P_1 and the top facet of P_2 are the same convex polygon. The maximum overlap is obtained by aligning the top facets of P_1 and P_2. The slab S consists of translations that place P_1 between the horizontal planes through the top facet and the bottom vertex of P_2. Thus, for any vertex of P_1 and any edge or facet of P_2, some translation in S bring them into intersection, implying that $\Theta(n^2)$ elements of Γ stab the slab S. To generalize the example to $\mathbb{R}^d$, one can replace the top facet of P_2 and the top and bottom facets of P_1 by the same simple convex polytope in $\mathbb{R}^{d-1}$. Then, the same reasoning shows that $\Theta(n^{2\lfloor d/2 \rfloor})$ elements of Γ stab S.

Instead of parallel slabs, we propose to prune $\mathrm{Arr}(\Gamma)$ using the ε-net theory (Lemma 1). First, we define a set $\widehat{\Gamma}$ of hyperplanes, each containing one element of Γ. We generate a random subset $\widehat{\mathcal{E}}_0 \subset \widehat{\Gamma}$ of size $O(n^{\lfloor d/2 \rfloor} \log n)$. The ε-net theory ensures that $O(n^{\lfloor d/2 \rfloor})$ hyperplanes in $\widehat{\Gamma}$ stab any d-simplex in a cell of $\mathrm{Arr}(\widehat{\mathcal{E}}_0)$ with high probability, in particular, the cell C that contains the solution of the maximum overlap problem. How do we locate C? As binary search no longer works, we instead construct a sequence of cuttings on $\widehat{\mathcal{E}}_0$ to prune the search space to C, or more precisely a d-simplex $\rho_0 \subseteq C$ containing the solution. During this pruning, we recursively solve instances of the maximum overlap problem for P_1 and P_2 with translations restricted to a hyperplane in $\widehat{\mathcal{E}}_0$ in order to tell which side of this hyperplane we should step into.

The challenge is to find the elements of Γ that stab ρ_0 so that we can search in ρ_0 via cuttings. For the direct extension of POLYGON to high dimensions, we would scan the faces of P_1 and P_2 in a direction orthogonal to the slabs to find the face pairs that induce the elements of Γ stabbing a particular slab. However, in our case scanning no longer works. We prove a characterization of the elements of Γ that stab ρ_0, which allows us to find them using linear programming on P_1 and P_2. This is the key idea to defy the $O(n^{2\lfloor d/2 \rfloor})$ bound. The speedup in $\mathbb{R}^3$ is obtained by replacing the linear programming with suitable queries using the Dobkin-Kirkpatrick structure [8].

Degeneracy in P_1, P_2 and $\mathrm{Arr}(\widehat{\Gamma})$ has a great impact on the running time. For efficiency, the linear programming step requires each face of P_1 and P_2 to be incident to $O(1)$ other faces. When pruning the search space using a cutting, we need to decide which side of a hyperplane $\ell \in \widehat{\mathcal{E}}_0$ to step into, after obtaining the translation $\alpha \in \ell$ that maximizes Q_α over ℓ. If α lies in a cell of $\mathrm{Arr}(\widehat{\Gamma})$ that is incident to many other cells, it may take a long time to decide which side of ℓ we should step into. This explains the need for the bounded incidence condition. (A precise definition is given in the next section.) We prove that the bounded incidence condition holds with probability one by perturbing P_1 and P_2 and using randomization to generate the hyperplanes in $\widehat{\Gamma}$. We can control the perturbation of P_1 and P_2 so that the maximum overlap decreases negligibly.

4 Algorithm

We first give some definitions and then elaborate on the algorithm outlined in the previous section. For each element $\gamma_{f,g} \in \Gamma$, define a hyperplane $\widehat{\gamma}_{f,g}$ containing $\gamma_{f,g}$ as follows.

- Suppose that $\dim(f) + \dim(g) < d$. If $\dim(\gamma_{f,g}) = d-1$, then $\widehat{\gamma}_{f,g} = \mathrm{aff}(\gamma_{f,g})$. Otherwise, we pick a unit vector v orthogonal to $\mathrm{aff}(\gamma_{f,g})$ uniformly at random and define $\widehat{\gamma}_{f,g}$ to be the $(d-1)$-flat through $\gamma_{f,g}$ orthogonal to v. That is, v is a random point on the unit sphere in the linear subspace of dimension $d - \dim(\gamma_{f,g})$ orthogonal to $\mathrm{aff}(\gamma_{f,g})$. Figure 3 shows some examples.

- Suppose that $\dim(f) + \dim(g) \geq d$. Since $\dim(\gamma_{f,g}) < d$ by the definition of Γ, there is a face h of f such that $\dim(h) + \dim(g) < d$ and $\mathrm{aff}(\gamma_{h,g}) = \mathrm{aff}(\gamma_{f,g})$. (Pick any if there are more than one such h's.) The hyperplane $\widehat{\gamma}_{h,g}$ is already defined in the previous case. We set $\widehat{\gamma}_{f,g} = \widehat{\gamma}_{h,g}$. Figure 4 shows an example.

We define $\widehat{\Gamma}$ to be the multiset $\{\, \widehat{\gamma}_{f,g} : \gamma_{f,g} \in \Gamma \,\}$. Duplicates exist in $\widehat{\Gamma}$ if two distinct face pairs induce the same hyperplane. Both $\widehat{\Gamma}$ and Γ have $O\big(n^{2\lfloor d/2 \rfloor}\big)$ elements, so we cannot afford to generate either of them completely.

Consider two quantities. The first one is the maximum number of faces in P_1 or P_2 that have a non-empty common intersection. The second one is the maximum number of hyperplanes in $\widehat{\Gamma}$ that have a non-empty common intersection. If these quantities have a constant upper bound,

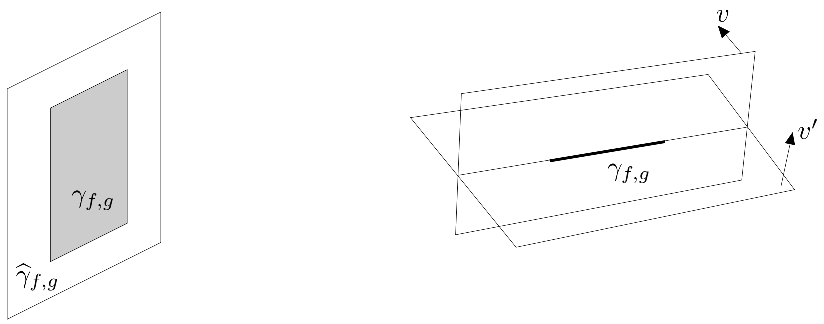

Figure 3: Illustrations of $\widehat{\gamma}_{f,g}$ in $\mathbb{R}^3$ when $\dim(f)+\dim(g) < 3$. Left: For a facet f and a vertex g, we have $\widehat{\gamma}_{f,g} = \text{aff}(\gamma_{f,g})$. Right: When f is an edge and g is a vertex, $\gamma_{f,g}$ is an open line segment and the plane $\widehat{\gamma}_{f,g}$ depends on the choice of the normal vector orthogonal to $\gamma_{f,g}$. Two possible vectors v and v' are shown with the corresponding planes containing $\gamma_{f,g}$.

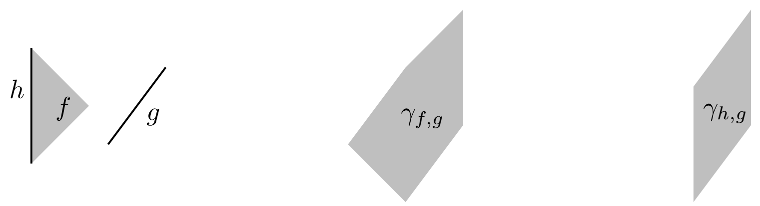

Figure 4: Illustration of $\widehat{\gamma}_{f,g}$ in $\mathbb{R}^3$ when $\dim(f) + \dim(g) \geq 3$. Left: The faces f of P_1 (a triangle) and g of P_2 (an edge) are parallel. So $\dim(\gamma_{f,g}) = 2$ and $\dim(f)+\dim(g) = 3$. Middle: The shaded polygon denotes $\gamma_{f,g}$, so $\widehat{\gamma}_{f,g}$ is the supporting plane of the polygon. Right: The shaded parallelogram is $\gamma_{h,g}$ which is coplanar with $\gamma_{f,g}$ as h is an edge of f. So $\dim(\gamma_{h,g}) = 2$ and $\dim(h) + \dim(g) < 3$. We can set $\widehat{\gamma}_{f,g}$ to be $\widehat{\gamma}_{h,g}$.

Locate(Π) /* return the optimal translation in Π */

1. If $\dim(\Pi) = 0$, return Π; otherwise, construct a d-simplex τ_0 that contains the optimal translation in Π.
2. Sample a subset $\widehat{\mathcal{E}}_0 \subset \widehat{\Gamma}$ of $\Theta(n^{\lfloor d/2 \rfloor} \log n)$ hyperplanes.
3. $\rho_0 := \text{Prune}(\Pi, \tau_0, \widehat{\mathcal{E}}_0)$.
4. Compute a subset $\widehat{\mathcal{E}}_1 \subset \widehat{\Gamma}$ that has $O(n^{\lfloor d/2 \rfloor})$ hyperplanes and contains $\{\widehat{\gamma}_{f,g} \in \widehat{\Gamma} : \gamma_{f,g} \cap \text{int}(\rho_0) \neq \emptyset\}$.
5. $\rho_1 := \text{Prune}(\Pi, \rho_0, \widehat{\mathcal{E}}_1)$.
6. Return the translation $\alpha \in \rho_1 \cap \Pi$ that maximizes $\text{vol}(Q_\alpha)^{1/d}$.

Prune($\Pi, \tau, \widehat{\mathcal{E}}$) /* return a d-simplex $\tau' \subseteq \tau$ such that τ' contains the optimal translation in Π and $\text{int}(\tau')$ lies in a cell of $\text{Arr}(\widehat{\mathcal{E}})$. */

1. Set $\tau' = \tau$. Let α denote the translation in Π that maximizes $\text{vol}(Q_\alpha)$ over Π.
2. Compute a $\frac{1}{2}$-cutting of $\widehat{\mathcal{E}}$. Find the d-simplex τ'' in the cutting that contains α.
3. Triangulate $\tau' \cap \tau''$. Update τ' to be the d-simplex in this triangulation that contains α. Remove from $\widehat{\mathcal{E}}$ the hyperplanes that avoid $\text{int}(\tau')$.
4. Return τ' if $\widehat{\mathcal{E}}$ becomes empty. Otherwise, go to step 2.

Figure 5: Pseudocodes of Locate and Prune.

the *bounded incidence condition* is satisfied. We assume this condition in the rest of the paper. The time complexity analysis of our algorithm depends on it although the correctness of our algorithm does not. We can show that the bounded incidence condition holds with probability one by perturbing P_1 and P_2 and using the randomization in the definitions of the hyperplanes in $\widehat{\Gamma}$. For any $\varepsilon > 0$, we can control the perturbation so that $\text{vol}(Q_{\alpha_\varepsilon})$ is at most ε less than the optimum, where α_ε is the translation that realizes the maximum overlap for the perturbed input. We give the proofs for this in the appendix in order to focus on the main algorithm.

We call our algorithm Locate. Given an m-flat Π, Locate(Π) returns the translation $\alpha \in \Pi$ that maximizes $\text{vol}(Q_\alpha)$ over Π. The original maximum overlap problem is solved by setting $m = d$. Locate calls a subroutine Prune that takes three parameters, an m-flat Π, a d-simplex τ containing the optimal translation in Π, and a subset $\widehat{\mathcal{E}} \subseteq \widehat{\Gamma}$. Prune($\Pi, \tau, \widehat{\mathcal{E}}$) outputs a d-simplex $\tau' \subseteq \tau$ such that τ' contains the optimal translation in Π and $\text{int}(\tau')$ lies in a cell of $\text{Arr}(\widehat{\mathcal{E}})$. Figure 5 shows the pseudocodes of Locate and Prune. Although the solution lies in the m-flat Π, we search the arrangement $\text{Arr}(\widehat{\Gamma})$ in $\mathbb{R}^d$ for notational convenience.

4.1 How Locate works

Refer to the pseudocode of Locate in Figure 5. In step 1, τ_0 is constructed as follows. For $j \in \{1, 2\}$, we compute P_j and its axes-parallel bounding box B_j in $O(n^{\lfloor d/2 \rfloor} + n \log n)$ time [5]. The translations that bring B_1 and B_2 into intersection form a box B which can be computed in $O(1)$ time. We can take τ_0 to be any d-simplex containing B. By steps 2 and 3, we call Prune$(\Pi, \tau_0, \widehat{\mathcal{E}}_0)$ with a random subset $\widehat{\mathcal{E}}_0 \subset \widehat{\Gamma}$. We want the d-simplex ρ_0 returned by Prune to be stabbed by only few hyperplanes in $\widehat{\Gamma}$ because we will construct cuttings on them later. By the ε-net theory, a d-simplex in any cell of $\mathrm{Arr}(\widehat{\mathcal{E}}_0)$ is stabbed by $(|\widehat{\Gamma}|/|\widehat{\mathcal{E}}_0|)\log n$ hyperplanes with probability $1 - n^{-O(1)}$. We have $|\widehat{\Gamma}| = O\big(n^{2\lfloor d/2 \rfloor}\big)$ and we make $|\widehat{\mathcal{E}}_0| = O(n^{\lfloor d/2 \rfloor}\log n)$ to optimize the running time of Locate. The lemma below explains how we pick a subset $\widehat{\mathcal{E}}_0$ of hyperplanes from $\widehat{\Gamma}$.

Lemma 3 *We can sample in $O(n^{\lfloor d/2 \rfloor}\log^2 n)$ time a subset $\widehat{\mathcal{E}}_0 \subset \widehat{\Gamma}$ of size $O(n^{\lfloor d/2 \rfloor}\log n)$ such that, with probability $1 - n^{-O(1)}$, for any d-simplex ρ whose interior lies in a cell of $\mathrm{Arr}(\widehat{\mathcal{E}}_0)$, only $O(n^{\lfloor d/2 \rfloor})$ hyperplanes in $\widehat{\Gamma}$ intersect $\mathrm{int}(\rho)$.*

Proof. Let F_j^i be the number of i-faces of P_j for $j \in \{1, 2\}$ and $i \in [0, d]$. For $k \in [0, d-1]$, let $\widehat{\Gamma}_k$ be the multiset $\{\, \widehat{\gamma}_{f,g} \in \widehat{\Gamma} : \dim(f) + \dim(g) = k \,\}$. We sample a hyperplane from $\widehat{\Gamma}_k$ uniformly at random as follows. First, pick an integer $i \in [0, k]$ with probability $F_1^i F_2^{k-i}/(\sum_{a=0}^{k} F_1^a F_2^{k-a})$. Second, pick an i-face of P_1 and a $(k-i)$-face of P_2 with probabilities $1/F_1^i$ and $1/F_2^{k-i}$, respectively. Repeat to pick $\Theta(n^{\lfloor d/2 \rfloor}\log n)$ face pairs that induce $\Theta(n^{\lfloor d/2 \rfloor}\log n)$ hyperplanes in $\widehat{\Gamma}_k$. The set $\widehat{\mathcal{E}}_0$ contains all hyperplanes sampled over $k \in [0, d-1]$ with duplicates removed via sorting. The time needed is $O(n^{\lfloor d/2 \rfloor}\log^2 n)$.

Take any d-simplex ρ whose interior lies in a cell of $\mathrm{Arr}(\widehat{\mathcal{E}}_0)$. It follows immediately from Lemma 1 that, with probability $1 - n^{-O(1)}$, only $O(n^{\lfloor d/2 \rfloor})$ hyperplanes in $\bigcup_{k=0}^{d-1}\widehat{\Gamma}_k$ intersects $\mathrm{int}(\rho)$. It is possible for $\mathrm{int}(\rho)$ to intersect a hyperplane $\widehat{\gamma}_{f,g}$ in $\widehat{\Gamma}$ where $\dim(f) + \dim(g) \geq d$ and so $\widehat{\gamma}_{f,g} \notin \bigcup_{k=0}^{d-1}\widehat{\Gamma}_k$. By the definition of $\widehat{\Gamma}$, we have $\widehat{\gamma}_{f,g} = \widehat{\gamma}_{h,g}$ for some face h of f where $\dim(h) + \dim(g) < d$, which implies that $\widehat{\gamma}_{h,g} \in \bigcup_{k=0}^{d-1}\widehat{\Gamma}_k$. We charge the intersection between $\mathrm{int}(\rho)$ and $\widehat{\gamma}_{f,g}$ to the intersection between $\mathrm{int}(\rho)$ and $\widehat{\gamma}_{h,g}$. By the bounded incidence condition, the intersection between $\mathrm{int}(\rho)$ and $\widehat{\gamma}_{h,g}$ is charged only $O(1)$ times. ◻

We discuss how Prune works in the next section, and we defer to Section 4.3 the discussion of step 4, the generation of a subset $\widehat{\mathcal{E}}_1 \subset \widehat{\Gamma}$ that contains $\{\widehat{\gamma}_{f,g} \in \widehat{\Gamma} : \gamma_{f,g} \cap \mathrm{int}(\rho_0) \neq \emptyset\}$. After step 5, we have a d-simplex ρ_1 such that ρ_1 contains the optimal translation in Π and $\mathrm{int}(\rho_1)$ lies in a cell of $\mathrm{Arr}(\widehat{\mathcal{E}}_1)$. The property of $\widehat{\mathcal{E}}_1$ implies that $\mathrm{int}(\rho_1)$ lies in a cell of $\mathrm{Arr}(\Gamma)$. (Some $\widehat{\gamma}_{f,g}$ in $\widehat{\Gamma}$ may intersect $\mathrm{int}(\rho_1)$, but $\gamma_{f,g}$ does not.) We describe below how to find the optimal translation in step 6.

We first obtain a formula φ for $\mathrm{vol}(Q_\alpha)$ for any $\alpha \in \mathrm{int}(\rho_1)$ by defining a *canonical triangulation* T_α of Q_α as follows. The canonical triangulations of the $(d-1)$-faces of Q_α are recursively defined. Then, fix a vertex q of Q_α and connect it to every simplex in $\mathrm{bd}(Q_\alpha)$ not incident to q to get T_α. If we have the volume formulae for the d-simplices in T_α, their sum gives the formula for $\mathrm{vol}(Q_\alpha)$. The *signed volume* of a d-simplex with vertices $v_0, v_1, \ldots, v_d$ is $\frac{1}{d!}\det(v_1 - v_0, v_2 - v_0, \ldots, v_d - v_0)$, where each v_i is viewed as a column vector. Since there is no combinatorial change as α varies in $\mathrm{int}(\rho_1)$, the vertex coordinates of Q_α are fixed linear functions in α and there is no combinatorial change in T_α. So the signed volumes of the d-simplices in T_α do not change sign. We construct T_{α_0} for a fixed translation $\alpha_0 \in \mathrm{int}(\rho_1)$ to determine which d-simplices in T_α have negative volumes and multiply their formulae by -1. Constructing Q_α and T_α takes $O(n^{\lfloor d/2 \rfloor} + n\log n)$ time and $|T_\alpha| = O(n^{\lfloor d/2 \rfloor})$. So we can compute a formula φ for $\mathrm{vol}(Q_\alpha)$ with $O(n^{\lfloor d/2 \rfloor})$ terms in $O(n^{\lfloor d/2 \rfloor} + n\log n)$ time.

Combinatorial changes may happen if we move α from $\mathrm{int}(\rho_1)$ to $\mathrm{bd}(\rho_1)$. Nonetheless, these possible changes are that some d-simplices in T_α may become degenerate and have zero volume. So the formula φ is valid for any $\alpha \in \rho_1$.

We convert φ to a formula ψ using the barycentric coordinates of $\alpha \in \rho_1 \cap \Pi$ as variables. The formula ψ has $O(n^{\lfloor d/2 \rfloor})$ terms and the conversion takes $O(n^{\lfloor d/2 \rfloor})$ time. We maximize

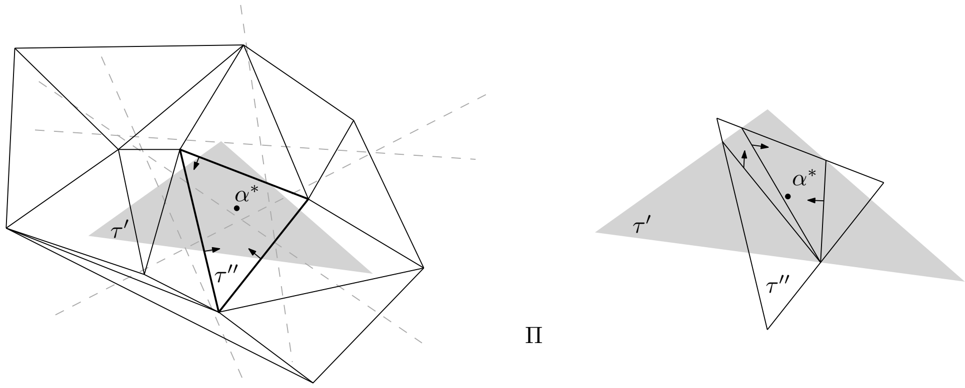

Figure 6: Illustrations of steps 2 and 3 of PRUNE in $\mathbb{R}^2$ when $\Pi = \mathbb{R}^2$. Left: The shaded triangle is τ'. The dashed lines form $\mathrm{Arr}(\widehat{\mathcal{E}})$ and the triangulation denotes a $\frac{1}{2}$-cutting of $\widehat{\mathcal{E}}$. We identify the triangle τ'' in the cutting (with bold edges) that contains α^*. Right: We intersect τ' with τ'' and triangulate $\tau' \cap \tau''$. Then, we find the triangle in the triangulation that contains α^*. This triangle becomes the new τ' for the next iteration of steps 2 and 3.

$\psi^{1/d}$ by standard calculus. If $\psi^{1/d}$ attains its maximum in $\mathrm{int}(\rho_1 \cap \Pi)$ (i.e., positive barycentric coordinates), we have the optimal translation. Otherwise, $\psi^{1/d}$ attains its maximum in $\mathrm{bd}(\rho_1 \cap \Pi)$ and we repeat the conversion of φ and the maximization for each face of $\rho_1 \cap \Pi$.

Lemma 4 LOCATE(Π) *runs in* $T(n,m) = T_g + T_p + O(n^{\lfloor d/2 \rfloor} \log^2 n)$ *time, where* T_g *denotes the time to generate* $\widehat{\mathcal{E}}_1$ *in step 4 and* T_p *denotes the total running time of* PRUNE *in steps 3 and 5.*

4.2 How PRUNE works

Let α^* denote the translation in an m-flat Π that maximizes the overlap over Π. PRUNE takes parameters Π, a d-simplex τ containing α^*, and a set $\widehat{\mathcal{E}}$ of hyperplanes. PRUNE returns a d-simplex $\tau' \subseteq \tau$ such that $\alpha^* \in \tau'$ and $\mathrm{int}(\tau')$ lies in a cell of $\mathrm{Arr}(\widehat{\mathcal{E}})$. Assume for now an oracle that, given any $(m-1)$-flat $\ell \subset \Pi$, decides which side of ℓ contains α^*.

Refer to the pseudocode of PRUNE in Figure 5. In step 2, we construct a $(1/2)$-cutting of $\widehat{\mathcal{E}}$ that has $O(1)$ size and can be computed in $O(|\widehat{\mathcal{E}}|)$ time by Lemma 2. Let H be the set of supporting hyperplanes of the $(d-1)$-simplices in the cutting. Running the oracle on $h \cap \Pi$ for all $h \in H$ tells us which sides of the hyperplanes in H contain α^*. This gives the d-simplex τ'' in the cutting that contains α^*. In step 3, we triangulate $\tau' \cap \tau''$ in $O(1)$ time and use the oracle as before to find the d-simplex in the triangulation that contains α^*. Figure 6 illustrates steps 2 and 3. By Lemma 2, at least half of the hyperplanes in $\widehat{\mathcal{E}}$ are removed in step 3. Thus, steps 2–4 iterate $O(\log |\widehat{\mathcal{E}}|)$ times and PRUNE takes $O(T_o \log |\widehat{\mathcal{E}}| + |\widehat{\mathcal{E}}| + \frac{1}{2}|\widehat{\mathcal{E}}| + \frac{1}{4}|\widehat{\mathcal{E}}| + \ldots) = O(T_o \log |\widehat{\mathcal{E}}| + |\widehat{\mathcal{E}}|)$ time, where T_o is the time to run the oracle once.

We describe below how the oracle works. Let F be the restriction of $\mathrm{vol}(Q_\alpha)^{1/d}$ to Π. For a cell C of $\mathrm{Arr}(\Gamma)$, let F_C denote the restriction of F to $\mathrm{cl}(C) \cap \Pi$ and let ∇F_C denote the gradient of F_C. We run LOCATE(ℓ) to find the translation $\tilde{\alpha} \in \ell$ that maximizes the overlap over ℓ. Intuitively, the gradient of F at $\tilde{\alpha}$ points to the side of ℓ containing α^*. However, this idea fails because F may not be smooth at $\tilde{\alpha}$, leaving the gradient of F undefined at $\tilde{\alpha}$. We get around this problem as follows. We call a cell C of $\mathrm{Arr}(\Gamma)$ *special* if $\mathrm{cl}(C)$ contains $\tilde{\alpha}$ and $\nabla F_C(\tilde{\alpha})$ points into C. If there is no special cell, we report that $\alpha^* = \tilde{\alpha}$. If there is a special cell C, we report the side of ℓ that $\nabla F_C(\tilde{\alpha})$ points to. We argue that our strategy is correct as follows. Take the path of steepest ascent on the graph of F from $F(\tilde{\alpha})$ to $F(\alpha^*)$ and project it to Π. If the projected path does not leave ℓ at $\tilde{\alpha}$, we have $\alpha^* = \tilde{\alpha}$, so for any cell C whose closure contains $\tilde{\alpha}$, the gradient $\nabla F_C(\tilde{\alpha})$ cannot point into C, i.e., no special cell. If the projected path leaves ℓ at $\tilde{\alpha}$, it enters a special cell C and, by the maximality of $F(\tilde{\alpha})$ over ℓ, the projected path never returns to ℓ. Thus, $\nabla F_C(\tilde{\alpha})$ points to the side of ℓ containing α^*. There cannot be two special cells. Otherwise, the steepest ascent at $F(\tilde{\alpha})$ projects to a direction v in Π that points outside some special cell C. By definition, $|\nabla F_C(\tilde{\alpha})|$ is greater than the magnitude of the gradient of F_C at $\tilde{\alpha}$ in direction v, which by the concavity of the graph of F, is at least the steepest ascent at $F(\tilde{\alpha})$. But then one can ascend faster on the graph of F_C in direction $\nabla F_C(\tilde{\alpha})$, a contradiction.

The oracle requires the computation of $\nabla F_C(\tilde{\alpha})$ for each cell C of $\mathrm{Arr}(\Gamma)$. We describe this computation in the following. Let $\mathcal{A} \subset \Gamma$ be the subset of elements whose closure contain $\tilde{\alpha}$. They are induced by the intersecting face pairs of $P_1 + \tilde{\alpha}$ and P_2, so we can compute $\mathcal{A}$ by constructing $Q_{\tilde{\alpha}}$ in $O(n^{\lfloor d/2 \rfloor} + n \log n)$ time. Let $\widehat{\mathcal{A}} = \{\widehat{\gamma}_{f,g} : \gamma_{f,g} \in \mathcal{A}\}$. We have $|\widehat{\mathcal{A}}| = O(1)$ by the bounded incidence condition as all hyperplanes in $\widehat{\mathcal{A}}$ go through $\tilde{\alpha}$. The closure of each cell of $\mathrm{Arr}(\widehat{\mathcal{A}})$ contains $\tilde{\alpha}$. Locally at $\tilde{\alpha}$, $\mathrm{Arr}(\widehat{\mathcal{A}})$ is a refinement of the cells of $\mathrm{Arr}(\Gamma)$ whose closure contain $\tilde{\alpha}$. So it suffices to compute $\nabla F_C(\tilde{\alpha})$ for each cell C of $\mathrm{Arr}(\widehat{\mathcal{A}})$, which can be done as follows. Compute the unit vector v that points into $\mathrm{cl}(C) \cap \Pi$ in the average direction of the edges of $\mathrm{cl}(C) \cap \Pi$. For any faces f of P_1 and g of P_2 where $(f + \tilde{\alpha}) \cap g \neq \emptyset$, we check whether $f + \tilde{\alpha} + rv$ intersects g, treating r as arbitrarily small. This gives the face lattice of $Q_{\tilde{\alpha}+rv}$. We want to compute the formula for $\mathrm{vol}(Q_{\tilde{\alpha}+rv})$ as in the previous section, but there is one difference. The face lattice of $Q_{\tilde{\alpha}+rv}$ allows us to construct the canonical triangulation $T_{\tilde{\alpha}+rv}$ of $Q_{\tilde{\alpha}+rv}$. This gives the signed volume formula for each d-simplex in $T_{\tilde{\alpha}+rv}$. The unknown r is the only variable in the formula. However, since we do not know an exact value of r, we cannot evaluate the signed volumes of the d-simplices in $T_{\tilde{\alpha}+rv}$ and flip the signs of the negative volumes in order to obtain a formula for $\mathrm{vol}(Q_{\tilde{\alpha}+rv})$. Instead, we decide whether a d-simplex τ in $T_{\tilde{\alpha}+rv}$ has negative volume as follows. Let V_τ denote the signed volume formula of τ, which is a polynomial in r of fixed degree. We compute the ith derivative $\frac{d^i V_\tau}{dr^i}$ for the smallest $i \geq 0$ such that $\frac{d^i V_\tau}{dr^i}|_{r=0}$ is non-zero. (The 0th derivative is V_τ itself.) If $\frac{d^i V_\tau}{dr^i}|_{r=0}$ is positive, then τ has positive volume; otherwise, τ has negative volume. This takes $O(1)$ time per d-simplex in $T_{\tilde{\alpha}+rv}$. Hence, for each cell C of $\mathrm{Arr}(\widehat{\mathcal{A}})$, we can compute $\nabla F_C(\tilde{\alpha})$ in $O(n^{\lfloor d/2 \rfloor} + n \log n)$ time.

Lemma 5 $\textsc{Prune}(\Pi, \tau, \widehat{\mathcal{E}})$ *runs in* $O(T(n, m-1) \log |\widehat{\mathcal{E}}| + n^{\lfloor d/2 \rfloor} \log |\widehat{\mathcal{E}}| + n \log n \log |\widehat{\mathcal{E}}| + |\widehat{\mathcal{E}}|)$ *time, where* $T(n, m-1)$ *is the time for* LOCATE *to run on an* $(m-1)$*-flat.*

4.3 The generation of $\widehat{\mathcal{E}}_1$

The step 4 of LOCATE generates a subset $\widehat{\mathcal{E}}_1 \subset \widehat{\Gamma}$ that contains the set $\{\, \widehat{\gamma}_{f,g} \in \widehat{\Gamma} : \gamma_{f,g} \cap \mathrm{int}(\rho_0) \neq \emptyset \,\}$. We discuss how to do this in $O(n^{\lfloor d/2 \rfloor + 1} \log n)$ time and ensure that $|\widehat{\mathcal{E}}_1| = O(n^{\lfloor d/2 \rfloor})$. Recall that the *Minkowski sum* of two subsets X and Y of $\mathbb{R}^d$ is $X \oplus Y = \{x + y : x \in X, y \in Y\}$. So $\dim(X \oplus Y) \leq \dim(X) + \dim(Y)$.

We first compute a set $\mathcal{E}_1$ of face pairs from P_1 and P_2 as follows, each inducing an element in Γ. We initialize $\mathcal{E}_1$ to be empty. For each face h_1 of P_1 and for each face σ of ρ_0, we compute the vertices of $(h_1 \oplus \sigma) \cap P_2$. For each vertex computed, if it is equal to $(\mathrm{int}(h_1) \oplus \sigma) \cap \mathrm{int}(h_2)$ for some face h_2 of P_2, we insert into $\mathcal{E}_1$ all face pairs (f, g) where $h_1 \in \mathrm{faces}(f)$ and $h_2 \in \mathrm{faces}(g)$ such that $\dim(\gamma_{f,g}) < d$. (By storing with f and g the basis vectors of $\mathrm{aff}(f)$ and $\mathrm{aff}(g)$, we can check in $O(1)$ time whether $\dim(f \oplus g) < d$ and this suffices as $\dim(\gamma_{f,g}) = \dim(f \oplus g)$.) The vertices of $(h_1 \oplus \sigma) \cap P_2$ that are not induced by $\mathrm{int}(h_1) \oplus \sigma$ do not trigger any insertion into $\mathcal{E}_1$. At the end, we set $\widehat{\mathcal{E}}_1 = \{\, \widehat{\gamma}_{f,g} : (f, g) \in \mathcal{E}_1 \,\}$ and remove the duplicates in $\widehat{\mathcal{E}}_1$ via sorting.

Our analysis in the rest of this section is divided into three parts. First, we show that $\widehat{\mathcal{E}}_1$ contains the set $\{\, \widehat{\gamma}_{f,g} \in \widehat{\Gamma} : \gamma_{f,g} \cap \mathrm{int}(\rho_0) \neq \emptyset \,\}$. Second, we show that $|\widehat{\mathcal{E}}_1| = O(n^{\lfloor d/2 \rfloor})$ with probability $1 - n^{-O(1)}$. Third, we show that, with probability $1 - n^{-O(1)}$, it takes $O(n^{\lfloor d/2 \rfloor + 1} \log n)$

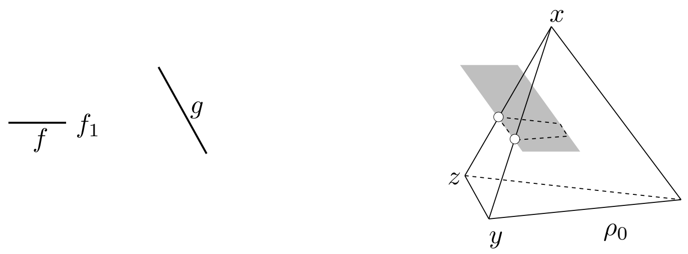

Figure 7: Illustration of the choice of γ_{h_1,h_2} and σ in $\mathbb{R}^3$. Left: An edge f of P_1 with an endpoint f_1 and an edge g of P_2. Right: The set $\mathrm{cl}(\gamma_{f,g})$ is the shaded parallelogram that intersects ρ_0. An edge of $\mathrm{cl}(\gamma_{f,g})$ corresponding to $\mathrm{cl}(\gamma_{f_1,g})$ is coplanar with the facet xyz of ρ_0 and $\mathrm{cl}(\gamma_{f_1,g})$ crosses xyz completely. We can choose $h_1 = f_1$ and $h_2 = g$ in this case. We can choose σ to be the edge xy or xz which intersects $\gamma_{f_1,g}$ in a single point.

time to compute the vertices of $(h_1 \oplus \sigma) \cap P_2$ over all faces h_1 of P_1 and all faces σ of ρ_0.

4.3.1 The first part

We first prove two geometric properties and then show that $\widehat{\mathcal{E}}_1$ contains the set $\{\,\widehat{\gamma}_{f,g} \in \widehat{\Gamma} : \gamma_{f,g} \cap \mathrm{int}(\rho_0) \neq \emptyset\,\}$.

Lemma 6 *The following properties hold for each element $\gamma_{f,g} \in \Gamma$.*

(i) *Suppose that $(\mathrm{int}(f) \oplus \sigma) \cap \mathrm{int}(g)$ is a single point for some face σ of ρ_0. Then, $\widehat{\gamma}_{f,g} \cap \mathrm{int}(\rho_0) \neq \emptyset$ or $\widehat{\gamma}_{f,g}$ contains a vertex of ρ_0.*

(ii) *Suppose that $\gamma_{f,g}$ intersects ρ_0. There exists a face h_1 of f, a face h_2 of g, and a face σ of ρ_0 such that $(\mathrm{int}(h_1) \oplus \sigma) \cap \mathrm{int}(h_2)$ is a single point.*

Proof. Since $(\mathrm{int}(f) \oplus \sigma) \cap \mathrm{int}(g) \neq \emptyset$, some translation in σ brings $\mathrm{int}(f)$ and $\mathrm{int}(g)$ into intersection. Thus, $\gamma_{f,g} \cap \sigma \neq \emptyset$ and (i) follows as σ is a face of ρ_0.

Consider (ii). Recall that $\gamma_{f,g}$ is a point or an open convex set. So $\mathrm{cl}(\gamma_{f,g})$ is a convex polytope. Among the faces of $\mathrm{cl}(\gamma_{f,g})$ that intersect ρ_0, we choose those with the lowest dimension. Among these faces, we choose a face $\mathrm{cl}(\gamma_{h_1,h_2})$ such that $\dim(h_1) + \dim(h_2)$ is minimum. Figure 7 gives an illustration. Since ρ_0 does not intersect any face of $\mathrm{cl}(\gamma_{f,g})$ with dimension less than $\dim(\gamma_{h_1,h_2})$, the boundary of $\mathrm{cl}(\gamma_{h_1,h_2})$ avoids ρ_0, which implies that some face σ of ρ_0 intersects γ_{h_1,h_2} in a single point. That is, there is a unique translation $\alpha = \gamma_{h_1,h_2} \cap \sigma$ such that $(\mathrm{int}(h_1) + \alpha) \cap \mathrm{int}(h_2) \neq \emptyset$. We claim that $(\mathrm{int}(h_1) + \alpha) \cap \mathrm{int}(h_2)$ is a single point, which implies (ii). If $(\mathrm{int}(h_1) + \alpha) \cap \mathrm{int}(h_2)$ is not a single point, its closure has a vertex $(\mathrm{int}(h_1') + \alpha) \cap \mathrm{int}(h_2')$ for some $h_1' \in \mathrm{faces}(h_1)$ and $h_2' \in \mathrm{faces}(h_2)$ where h_1' is a proper face of h_1 or h_2' is a proper face of h_2. Thus, $\dim(\gamma_{h_1',h_2'}) \leq \dim(\gamma_{h_1,h_2})$ and $\gamma_{h_1',h_2'}$ intersects σ, but $\dim(h_1') + \dim(h_2') < \dim(h_1) + \dim(h_2)$. This contradicts our choice of γ_{h_1,h_2}. ◻

Lemma 7 *$\widehat{\mathcal{E}}_1$ contains the set $\{\,\widehat{\gamma}_{f,g} \in \widehat{\Gamma} : \gamma_{f,g} \cap \mathrm{int}(\rho_0) \neq \emptyset\,\}$.*

Proof. Take an element $\gamma_{f,g}$ of Γ that intersects $\mathrm{int}(\rho_0)$. By Lemma 6(ii), $(\mathrm{int}(h_1) \oplus \sigma) \cap \mathrm{int}(h_2)$ is a single point for a face h_1 of f, a face h_2 of g, and a face σ of ρ_0. So $(\mathrm{int}(h_1) \oplus \sigma) \cap \mathrm{int}(h_2)$ is a vertex of $(h_1 \oplus \sigma) \cap P_2$. Our procedure collects this vertex and adds (f, g) to $\mathcal{E}_1$. ◻

4.3.2 The second part

The lemma below gives an $O(n^{\lfloor d/2 \rfloor})$ bound on the number of vertices computed by our generation procedure. It follows that $|\widehat{\mathcal{E}}_1| = O(n^{\lfloor d/2 \rfloor})$.

Lemma 8 *With probability $1 - n^{-O(1)}$, there are $O(n^{\lfloor d/2 \rfloor})$ vertices in the convex polytopes $(h_1 \oplus \sigma) \cap P_2$ over all faces h_1 of P_1 and all faces σ of ρ_0.*

Proof. Each vertex is $(\mathrm{int}(f) \oplus \sigma) \cap \mathrm{int}(g)$ for a face f of P_1, a face g of P_2, and a face σ of ρ_0. If $\dim(\gamma_{f,g}) < d$, we give the vertex a blue color; if $\dim(\gamma_{f,g}) = d$, we give it a red color. It is possible for a vertex to receive both colors if it is induced by two face pairs (f, g) and (f', g') such that $\dim(\gamma_{f,g}) < d$ and $\dim(\gamma_{f',g'}) = d$. We count these two colored instances of the same vertex separately in our analysis.

Consider the blue vertices. Lemma 6(i) implies that $\widehat{\gamma}_{f,g} \cap \mathrm{int}(\rho_0) \neq \emptyset$ or $\widehat{\gamma}_{f,g}$ contains a vertex of ρ_0. By Lemma 3, with probability $1 - n^{-O(1)}$, there are $O(n^{\lfloor d/2 \rfloor})$ hyperplanes $\widehat{\gamma}_{f,g}$ in $\widehat{\Gamma}$ where $\widehat{\gamma}_{f,g} \cap \mathrm{int}(\rho_0) \neq \emptyset$. By the bounded incidence condition, any vertex of ρ_0 lies in $O(1)$ hyperplanes $\widehat{\gamma}_{f,g}$ in $\widehat{\Gamma}$. For a face σ of ρ_0, the blue vertex $(\mathrm{int}(f) \oplus \sigma) \cap \mathrm{int}(g)$ may be constructed more than once if there are other faces $f' \in \mathrm{faces}(P_1)$ and $g' \in \mathrm{faces}(P_2)$ such that $(\mathrm{int}(f') \oplus \sigma) \cap \mathrm{int}(g') = (\mathrm{int}(f) \oplus \sigma) \cap \mathrm{int}(g)$. In this case, $f \cap f' \neq \emptyset$ and $g \cap g' \neq \emptyset$, which implies that the blue vertex $(\mathrm{int}(f) \oplus \sigma) \cap \mathrm{int}(g)$ is generated only $O(1)$ times by the bounded incidence condition. Another factor $2^{d+1} - 1$ is needed as we go over all faces σ of ρ_0. So we compute $O(n^{\lfloor d/2 \rfloor})$ blue vertices, counting multiplicities.

Consider a red vertex $(\mathrm{int}(f) \oplus \sigma) \cap \mathrm{int}(g)$. For any translation $\alpha \in \sigma$, we have $(\mathrm{int}(f) + \alpha) \cap \mathrm{int}(g) \subseteq (\mathrm{int}(f) \oplus \sigma) \cap \mathrm{int}(g)$, which is a single point. Therefore, for any translation $\alpha \in \sigma$, if $(\mathrm{int}(f) + \alpha) \cap \mathrm{int}(g) \neq \emptyset$, then

$$(\mathrm{int}(f) \oplus \sigma) \cap \mathrm{int}(g) \;=\; (\mathrm{int}(f) + \alpha) \cap \mathrm{int}(g). \tag{1}$$

Fix σ and a translation α_0 in σ. Divide the red vertices $(\mathrm{int}(f) \oplus \sigma) \cap \mathrm{int}(g)$ over all faces f of P_1 and g of P_2 into two groups, one satisfying $(\mathrm{int}(f) + \alpha_0) \cap \mathrm{int}(g) \neq \emptyset$ and the other satisfying $(\mathrm{int}(f) + \alpha_0) \cap \mathrm{int}(g) = \emptyset$. By (1), the number of red vertices in the first group is no more than the number of vertices of $(P_1 + \alpha_0) \cap P_2$, which is $O(n^{\lfloor d/2 \rfloor})$. For each red vertex $(\mathrm{int}(f) \oplus \sigma) \cap \mathrm{int}(g)$ in the second group, we charge it to a blue vertex as follows. Since $(\mathrm{int}(f) \oplus \sigma) \cap \mathrm{int}(g)$ is a single point, by continuity, $(f \oplus \sigma) \cap g$ is equal to this single point. We choose a face h_1 of f and a face h_2 of g such that $(\mathrm{int}(h_1) \oplus \sigma) \cap \mathrm{int}(h_2) \neq \emptyset$ and $\dim(h_1) + \dim(h_2)$ is minimized. Thus, $(\mathrm{int}(h_1) \oplus \sigma) \cap \mathrm{int}(h_2)$ is the single point $(f \oplus \sigma) \cap g$ and the minimization ensures that $\dim(h_1) + \dim(h_2) < d$. Figure 8 gives an example. So $\dim(\gamma_{h_1,h_2}) \leq \dim(h_1) + \dim(h_2) < d$. It follows that $(\mathrm{int}(h_1) \oplus \sigma) \cap \mathrm{int}(h_2)$ is a blue vertex (it is a vertex of $(h_1 \oplus \sigma) \cap P_2$). We charge the red vertex $(\mathrm{int}(f) \oplus \sigma) \cap \mathrm{int}(g)$ to it. For another red vertex $(\mathrm{int}(f') \oplus \sigma) \cap \mathrm{int}(g')$ to charge to $(\mathrm{int}(h_1) \oplus \sigma) \cap \mathrm{int}(h_2)$, we must have $h_1 \in \mathrm{faces}(f')$ and $h_2 \in \mathrm{faces}(g')$. So the blue vertex $(\mathrm{int}(h_1) \oplus \sigma) \cap \mathrm{int}(h_2)$ is charged $O(1)$ times by the bounded incidence condition. It follows that $O(n^{\lfloor d/2 \rfloor})$ red vertices are induced by each face σ of ρ_0. Another factor $2^{d+1} - 1$ is needed as we go over all faces σ of ρ_0. Like blue vertices, a red vertex may be generated more than once, but only $O(1)$ times. Thus, $O(n^{\lfloor d/2 \rfloor})$ red vertices are computed, counting multiplicities. ◻

4.3.3 The third part

The next result bounds the time to generate $\widehat{\mathcal{E}}_1$.

Lemma 9 *Computing $\widehat{\mathcal{E}}_1$ takes $O(n^{\lfloor d/2 \rfloor + 1} \log n)$ time with probability $1 - n^{-O(1)}$.*

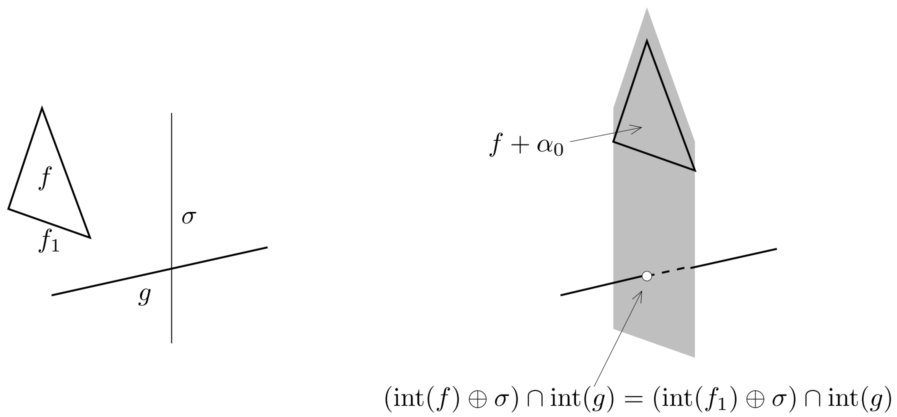

Figure 8: Illustration of a red vertex $(\mathrm{int}(f) \oplus \sigma) \cap \mathrm{int}(g)$ and the choice of γ_{h_1,h_2} in $\mathbb{R}^3$. Left: The vertical triangle is f and the edge is g. The bottom edge of f is f_1. The face σ of ρ_0 is the vertical edge shown. Right: The shaded polygon is $f \oplus \sigma$ and its interior is $\mathrm{int}(f) \oplus \sigma$. The intersection $(f \oplus \sigma) \cap g = (\mathrm{int}(f) \oplus \sigma) \cap \mathrm{int}(g)$ is a single point. Not all translations in σ bring f and g into intersection; for example, the translate $f + \alpha_0$ shown avoids g. Notice that $\mathrm{int}(f_1) \oplus \sigma$ intersects $\mathrm{int}(g)$ at the same intersection point $(f \oplus \sigma) \cap g$. We can choose $h_1 = f_1$ and $h_2 = g$. The faces f_1 and g induce a blue vertex as $\dim(f_1) + \dim(g) < 3$.

Proof. Let h_1 be a face of P_1 and let σ be a face of ρ_0. The face h_1 is the intersection of $O(n)$ halfspaces and hyperplanes. The Minkowski sum of each such halfspace or hyperplane with σ has $O(1)$ size and can be computed in $O(1)$ time. So the linear constraints defining $h_1 \oplus \sigma$ can be computed in $O(n)$ time.

We run Megiddo's linear programming algorithm to find a vertex ν of $(h_1 \oplus \sigma) \cap P_2$ in $O(n)$ time [14]. We visit the vertices adjacent to ν in two steps. First, we compute the supporting lines of edges incident to ν as follows. The point ν is dual to a $(d-1)$-flat and each bounding hyperplane through ν is dual to a point in this $(d-1)$-flat. The supporting lines of the edges incident to ν correspond to the $(d-2)$-faces of the convex hull of the dual points. By the bounded incidence condition and the constant size of σ, there are $O(1)$ such dual points, so it takes $O(1)$ time to compute their convex hull and hence the supporting lines of the edges incident to ν. Second, we shoot rays from ν along all these supporting lines and find the first hyperplane that each ray stops at by checking the linear constraints not containing ν in $O(n)$ time. These stopping points are the vertices adjacent to ν. Altogether, we can visit the vertices adjacent to ν in $O(n)$ time. Hence, it takes $O(n + k_{\sigma,h_1} n \log n)$ time to visit all vertices of $(h_1 \oplus \sigma) \cap P_2$, where k_{σ,h_1} is the number of such vertices and the $O(\log n)$ term comes from using a dictionary to record the vertices visited.

A vertex of $(h_1 \oplus \sigma) \cap P_2$ is equal to $(\mathrm{int}(h_1) \oplus \sigma) \cap \mathrm{int}(h_2)$ for some face h_2 of P_2 if and only if that vertex lies in the translates of the bounding hyperplanes through h_1 but not in the translate of any other bounding hyperplane of P_1. These vertices can be recognized in $O(n)$ time each.

Hence, it takes $O(n^{\lfloor d/2 \rfloor + 1} \log n)$ time to construct $\widehat{\mathcal{E}}_1$ because P_1 has $O(n^{\lfloor d/2 \rfloor})$ faces and $\sum_{\sigma,h_1} k_{\sigma,h_1} = O(n^{\lfloor d/2 \rfloor})$ with probability $1 - n^{-O(1)}$ by Lemma 8. (We also need to remove duplicates in $\widehat{\mathcal{E}}_1$ in $O(n^{\lfloor d/2 \rfloor} \log n)$ time via sorting.) ◻

By the results in Lemmas 4, 5, 8, and 9, we have the recurrence $T(n, m) = O(T(n, m - 1) \log n + n^{\lfloor d/2 \rfloor + 1} \log n)$ with boundary condition $T(n, 0) = O(1)$. The solution is $T(n, m) = O(mn^{\lfloor d/2 \rfloor + 1} \log^m n)$.

Theorem 1 *Let P_1 and P_2 be two convex polytopes in $\mathbb{R}^d$, $d \geq 3$, specified by n bounding hyperplanes. For any $\varepsilon > 0$, we can compute an overlap of P_1 and P_2 under translation that is at most ε less than the optimum. The running time is $O(n^{\lfloor d/2 \rfloor + 1} \log^d n)$ with probability $1 - n^{-O(1)}$.*

5 A faster algorithm for three dimensions

To obtain a better running time in $\mathbb{R}^3$, some changes are needed in step 2 of Locate and Lemmas 3, 4, 5, 8, and 9.

First, we decrease the size of the sample $\widehat{\mathcal{E}}_0$ from $\Theta(n \log n)$ to $\Theta(n\sqrt{\log n})$. So the time needed to sample $\widehat{\mathcal{E}}_0$ in Lemma 3 is improved to $O(n \log^{1.5} n)$. The running time of Locate in Lemma 4 is improved to $O(T_g + T_p + n \log^{1.5} n)$, where T_g is the time to generate $\widehat{\mathcal{E}}_1$ and T_p is the time to call Prune in steps 3 and 5.

In the proof of Lemma 5, we show that a halfspace can be eliminated from Π in $T(n, m-1) + O(n \log n)$ time. The $O(n \log n)$ term stems from intersecting a translate of P_1 with P_2 using Chazelle's convex hull algorithm in $\mathbb{R}^d$ for $d \geq 3$ [5]. In $\mathbb{R}^3$, Chazelle gave an $O(n)$-time algorithm to intersect two convex polyhedra [4]. Thus, the running time of Prune$(\Pi, \tau, \widehat{\mathcal{E}})$ in Lemma 5 can be improved to $O(T(n, m-1) \log |\widehat{\mathcal{E}}| + n \log |\widehat{\mathcal{E}}| + |\widehat{\mathcal{E}}|)$.

Since we change the size of $\widehat{\mathcal{E}}_0$ from $\Theta(n \log n)$ to $\Theta(n\sqrt{\log n})$, by Lemma 1, the bound in Lemma 8 on the number of vertices generated changes from $O(n)$ to $O(n\sqrt{\log n})$. This also implies that $|\widehat{\mathcal{E}}_1| = O(n\sqrt{\log n})$. Therefore, the time T_p to call Prune in steps 3 and 5 of Locate is now $O((T(n, m-1) \log n + n \log n)$.

It remains to show a faster method to generate $\widehat{\mathcal{E}}_1$. We speed up the generation of the set $\widehat{\mathcal{E}}_1$ of hyperplanes using the Dobkin-Kirkpatrick structure (DK-structure for short) [8]. Intuitively, the DK-structure allows us to visit a vertex of $(f \oplus \sigma) \cap P_2$ in $O(\log n)$ time for all faces f of P_1 and all faces σ of ρ_0, so that $\widehat{\mathcal{E}}_1$ can be generated in $O(n \log^{1.5} n)$ time. (Details will be given shortly.)

The new recurrence becomes:

$$\begin{aligned} T(n, m) &= O(T(n, m-1) \log n + n \log^{1.5} n) \\ T(n, 0) &= O(1). \end{aligned}$$

Hence, $T(n, m) = O(mn \log^{m+1/2} n)$, implying that the maximum overlap of P_1 and P_2 can be computed in $O(n \log^{3.5} n)$ time with high probability.

We elaborate on the generation of $\widehat{\mathcal{E}}_1$. Let f be a face of P_1 and let n_f denote the complexity of f. Let σ be a face of ρ_0. Since σ has constant size, we can compute the Minkowski sum $f \oplus \sigma$ in $O(n_f)$ time. This is a convex polyhedron with $O(n_f)$ size, possibly degenerated to a convex polygon, a line segment, or a point. We build the DK-structure for $f \oplus \sigma$ in $O(n_f \log n_f)$ time and the DK-structure for P_2 in $O(n \log n)$ time [8]. The structures support the following operations:

- Given a line ℓ, decide if ℓ intersects P_2 and if so, report the faces of P_2 that ℓ intersects. There are at most two intersection points. The query time is $O(\log n)$.

- Given a facet β of $f \oplus \sigma$, decide if β intersects P_2 and if so, report a point x in $\beta \cap P_2$. The query time is $O(\log n + \log n_f)$.

- Given a ray shooting from a point in $\mathrm{int}(P_2)$, return the face of P_2 hit by the ray. The query time is $O(\log n)$. The same can be done for a facet of P_2 or $f \oplus \sigma$ in $O(\log n)$ or $O(\log n_f)$ time, respectively.

We describe how to generate the vertices of $(f \oplus \sigma) \cap P_2$ in $O(n_f \log n + v_{\sigma,f} \log n)$ time, where $v_{\sigma,f}$ stands for the number of such vertices. These vertices fall into four categories and we discuss how to find them in each case.

Category 1: The intersections between the edges of $f \oplus \sigma$ and the facets of P_2. For each edge e of $f \oplus \sigma$, we take the supporting line ℓ of e and query the DK-structure to find the intersections between ℓ and the boundary of P_2 in $O(\log n)$ time. We report those intersections that lie on e. Therefore, it takes $O(n_f \log n)$ time to find the intersections between the edges of $f \oplus \sigma$ and the facets of P_2.

Category 2: The intersections between the facets of $f \oplus \sigma$ and the edges of P_2. Take a facet β of $f \oplus \sigma$. We query the DK-structure to find a point x in $\beta \cap P_2$. If no such point is returned, $\beta \cap P_2 = \emptyset$; otherwise, we shoot a ray from x in β to hit a facet of P_2 in $O(\log n)$ time. This gives us a starting point to trace the boundary of $\beta \cap P_2$, which consists of one closed convex chain or a collection of open convex chain(s). In the latter case, each chain endpoint is a vertex of Category 1 and they have already been computed. Every chain edge is equal to $\beta \cap \beta'$ for some facet β' of P_2. To trace $\beta \cap \beta'$, we shoot a ray along $\beta \cap \beta'$; if the ray hits $\mathrm{bd}(\beta')$ before $\mathrm{bd}(\beta)$, we find a vertex of $\beta \cap P_2$ to be reported; if the ray hits $\mathrm{bd}(\beta)$ before $\mathrm{bd}(\beta')$, we have reached an endpoint of a chain. This takes $O(\log n + \log n_f) = O(\log n)$ time per chain edge.

Category 3: The vertices of P_2 in $f \oplus \sigma$. We find these vertices of P_2 by tracing the edges of P_2 clipped inside $f \oplus \sigma$. The edges of P_2 are clipped exactly at vertices of Category 2, which have already been computed. So we can trace the edges of P_2 clipped inside $f \oplus \sigma$ in linear time. It is possible that P_2 lies inside $f \oplus \sigma$ in which case there is no vertex of Category 2. Then, we take a vertex x of P_2 and test in $O(n_f)$ time whether x lies inside $f \oplus \sigma$. If so, all vertices of P_2 lie inside $f \oplus \sigma$; otherwise, no vertex of P_2 lies inside $f \oplus \sigma$.

Category 4: The vertices of $f \oplus \sigma$ in P_2. These vertices can be determined in almost the same way as the vertices of Category 3 because the edges of $f \oplus \sigma$ are clipped at vertices of Category 1. The difference lies in testing whether a vertex x of $f \oplus \sigma$ lies in P_2 when there is no vertex of Category 1. We take the supporting line ℓ of any edge incident to x and query the DK-structure to find the intersections between ℓ and $\mathrm{bd}(P_2)$ in $O(\log n)$ time. If x lies between these intersections, then x lies in P_2 and so do other vertices of $f \oplus \sigma$. Otherwise, no vertex of $f \oplus \sigma$ lies in P_2.

The construction of the DK-structures takes $O(n \log n + \sum_{\sigma,f} n_f \log n_f) = O(n \log n)$ time. The time spent on identifying the vertices of $(f \oplus \sigma) \cap P_2$ over all $f \in \mathrm{faces}(P_1)$ and all $\sigma \in \mathrm{faces}(\rho_0)$ is $O(\sum_{\sigma,f} n_f \log n + \sum_{\sigma,f} v_{\sigma,f} \log n) = O(n \log^{1.5} n)$ time because the bound on $\sum_{\sigma,f} v_{\sigma,f}$ in Lemma 8 has changed in the 3D case from $O(n)$ to $O(n\sqrt{\log n})$.

Theorem 2 *Let P_1 and P_2 be two convex polyhedra in $\mathbb{R}^3$ specified by n planes. For any $\varepsilon > 0$, we can compute an overlap of P_1 and P_2 under translation that is at most ε less than the optimum. The running time is $O(n \log^{3.5} n)$ with probability $1 - n^{-O(1)}$.*

6 Discussion

The additive error ε is introduced because we perturb the input to improve the time complexities. It would be interesting to study if the perturbation can be removed in order to remove the additive error. Our running time of $O(n^{\lfloor d/2 \rfloor + 1} \log^d n)$ is close to the worst-case complexity $\Theta(n^{\lfloor d/2 \rfloor})$ of a convex polytope. Can this gap be closed? It would also be interesting to find the maximum overlap or approximate maximum overlap under rigid motion efficiently.

References

[1] H.-K. Ahn, P. Brass, and C.-S. Shin. Maximum overlap and minimum convex hull of two convex polyhedra under translation. *Comput. Geom. Theory and Appl.*, 40 (2008), 171–177.

[2] H.-K. Ahn, O. Cheong, C.-D. Park, C.-S. Shin, and A. Vigneron. Maximizing the Overlap of Two Planar Convex Sets under Rigid Motions. *Comput. Geom. Theory and Appl.*, 37 (2007), 3–15.

[3] M. de Berg, O. Cheong, O. Devillers, M. van Kreveld, and M. Teillaud. Computing the Maximum Overlap of Two Convex Polygons under Translations. *Theory of Comput. Syst.*, 31 (1998), 613–628.

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

[5] B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discr. Comput. Geom.*, 9 (1993), 377–409.

[6] B. Chazelle. Cutting Hyperplanes for Divide-and-Conquer. *Discr. Comput. Geom.*, 9 (1993), 145–159.

[7] B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. *J. Alg.*, 21 (1996), 579–597.

[8] D. P. Dobkin and D. G. Kirkpatrick. Determining the separation of preprocessed polyhedra – a unified approach. *Proc. 17th Internat. Colloq. Automata Lang. Program.*, 1990, 400–413.

[9] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer-Verlag, 1987.

[10] K. Fukuda and T. Uno. Polynomial time algorithms for maximizing the intersection volume of polytopes. *Pacific J. Optimization*, 3 (2007), 37–52.

[11] D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. *Discr. Comput. Geom.*, 2 (1987), 127–151.

[12] H.J.A.M. Heijmans and A.V. Tuzikov. Similarity and symmetry measures for convex shapes using Minkowski addition. *IEEE Trans. PAMI*, 20 (1998), 980–993.

[13] V. Koltun. Lectures on *Advanced Geometric algorithms*, Spring, 2006.

[14] N. Megiddo. Linear programming in linear time when the dimension is fixed. *J. ACM*, 31 (1984), 114–127.

[15] F. Meyer and P. Bouthemy. Region-based tracking in an image sequence. *Proc. 2nd European Conf. Computer Vision*, 1992, 476–484.

[16] D.M. Mount, R. Silverman, and A.Y. Wu. On the area of overlap of translated polygons. *Computer Vision and Image Understanding*, 64 (1996), 53–61.

[17] J.R. Sangwine-Yager. Mixed Volumes. *Handbook on Convex Geometry, Vol. A*, (eds. P.M. Gruber and J.M. Wills), Elsevier, 1993, 43–71.

[18] R.C. Veltkamp and M. Hagedoorn. State of the art in shape matching. *Principles of Visual Information Retrieval*, (ed. M. Lew), Springer, 2001, 87–119.

[19] A. Vigneron. Geometric optimization and sums of algebraic functions. *Proc. ACM–SIAM Sympos. Alg.*, 2010, 906–917.

APPENDIX

A Input perturbation

A.1 The process

We construct the faces of P_j for $j \in \{1,2\}$ in $O(n^{\lfloor d/2 \rfloor} + n \log n)$ time to find the bounding hyperplanes that bound the $(d-1)$-faces in P_j. We use H_j to denote this set of hyperplanes of P_j. We use $\mathbf{n}_h$ to denote the unit outward normal of a bounding hyperplane h.

We define two angles θ and ψ to limit the perturbation magnitude as follows. Each boundary face f of P_j is contained in some bounding hyperplanes $h_1, \ldots h_i$ for some $i \geq 1$. We insert the unit outward normals $\mathbf{n}_{h_1}, \ldots, \mathbf{n}_{h_i}$ as point sites on the unit $(d-1)$-sphere $\mathbb{S}^{d-1}$. These point sites lie strictly inside one half of $\mathbb{S}^{d-1}$. A hyperplane intersecting $\mathbb{S}^{d-1}$ cuts it into two subsets, and we call the smaller of the two a cap. We compute the smallest cap that contains $\mathbf{n}_{h_1}, \ldots, \mathbf{n}_{h_i}$. This is a LP-type problem and it can be solved in deterministic linear time using the algorithm of Chazelle and Matoušek [7]. Let ψ_f be the angular radius of this smallest cap, which is less than $\pi/2$. Define $\psi = \max\{\, \psi_f : f \in \text{faces}(P_1) \cup \text{faces}(P_2),\ \dim(f) < d \,\}$, which is less than $\pi/2$. Let θ be some angle between 0 and $\arcsin(\frac{1}{2}\cos\psi)$. We will discuss the setting of θ in the proof of Lemma 11.

We perturb each hyperplane $h \in H_1 \cup H_2$ as follows:

1. Let f be the $(d-1)$-face supported by h. Draw a random *anchor point* $a_h = \sum_{i=1}^{k} \lambda_i v_i$ from f, where $v_1, \ldots, v_k$ are the vertices of f, by picking $\mu_1, \ldots, \mu_k$ from the range $[0,1]$ independently and uniformly at random and setting $\lambda_i = \mu_i / (\sum_{j=1}^{k} \mu_j)$.

2. Pick a unit outward normal $\mathbf{n}_{\tilde{h}}$ uniformly at random from the set $\{\, v \in \mathbb{S}^{d-1} : \angle v, \mathbf{n}_h \leq \theta \,\}$.

3. Let $\tilde{h}$ be the hyperplane through the anchor point a_h and orthogonal to $\mathbf{n}_{\tilde{h}}$.

Let $\tilde{H}_j$ denote the set $\{\tilde{h} : h \in H_j\}$. Each $\tilde{h} \in \tilde{H}_j$ delimits a bounding halfspace that $\mathbf{n}_{\tilde{h}}$ points away from. The common intersection of these bounding halfspaces is the perturbed polytope $\tilde{P}_j$ approximating P_j.

The construction of P_1 and P_2 can be done in $O(n^{\lfloor d/2 \rfloor} + n \log n)$ time. Afterwards, it takes time linear in the complexities of P_1 and P_2, which is $O(n^{\lfloor d/2 \rfloor})$, to execute the remaining steps to define θ and ψ and to perturb the hyperplanes in $H_1 \cup H_2$.

A.2 Additive error

Let D be the maximum diameter of the bounding boxes of P_1 and P_2. We can compute D in $O(n^{\lfloor d/2 \rfloor})$ time from the vertices of P_1 and P_2. (We can also work with the maximum diameter of P_1 and P_2 but this is a harder computation.) We first bound the directed Hausdorff distance from $\text{bd}(\tilde{P}_j)$ to $\text{bd}(P_j)$.

Lemma 10 *For $j \in \{1,2\}$, any point in $\text{bd}(\tilde{P}_j)$ is at distance $2D\sin\theta/\cos\psi$ or less from $\text{bd}(P_j)$.*

Proof. Let x be a point in $\text{bd}(\tilde{P}_j)$. Let $\tilde{h}$ be the bounding hyperplane that contains x.

Suppose that $x \in P_j$. So $\|x - a_h\| \leq D$. Let y be the orthogonal projection of x onto the hyperplane h. Either $y \in \text{bd}(P_j)$ or $y \notin P_j$. Since $x \in P_j$, the boundary of P_j must intersect xy. Thus, the distance between x and $\text{bd}(P_j)$ is at most $\|x - y\| \leq \|x - a_h\| \cdot \sin\theta \leq D\sin\theta$.

Suppose that $x \notin P_j$. Let z be the closest point in $\text{bd}(P_j)$ to x. Take a face f of P_j that has the lowest dimension among those containing z. So the vector $x - z$ lies in the convex cone spanned by the outward normals of the bounding hyperplanes containing f. Among these bounding hyperplanes, let h_1 be the one that minimizes $\angle(x-z), \mathbf{n}_{h_1}$. So $\angle(x-z), \mathbf{n}_{h_1} \leq \psi_f \leq \psi$. Because h_1 separates x and P_j but $\tilde{h}_1$ does not separate x and $\tilde{P}_j$, when we perturb h_1 to $\tilde{h}_1$, we sweep over x. Thus, xa_{h_1} makes an angle at most θ with h_1, so the distance between x and h_1 is at most $\|x - a_{h_1}\| \cdot \sin\theta \leq \|x-z\| \cdot \sin\theta + \|z - a_{h_1}\| \cdot \sin\theta \leq \|x-z\| \cdot \sin\theta + D\sin\theta$. Also, the distance between x and h_1 is at least $\|x-z\| \cdot \cos(\angle(x-z), \mathbf{n}_{h_1}) \geq \|x-z\| \cdot \cos\psi$. Thus, $\|x-z\| \cdot \cos\psi \leq \|x-z\| \cdot \sin\theta + D\sin\theta$, which implies that $\|x-z\| \leq D\sin\theta/(\cos\psi - \sin\theta) \leq 2D\sin\theta/\cos\psi$ because $\theta \leq \arcsin(\frac{1}{2}\cos\psi)$ by definition. $\square$

Next, we show that the additive error can be made ε or less for any $\varepsilon > 0$ by adjusting θ.

Lemma 11 *Let* $\texttt{opt}$ *be the maximum overlap of* P_1 *and* P_2. *For any* $\varepsilon > 0$, *we can compute a threshold* θ_ε *in* $O(n^{\lfloor d/2 \rfloor})$ *time such that, whenever* $\theta \leq \theta_\varepsilon$, *we have* $\mathrm{vol}((P_1 + \alpha_\theta) \cap P_2) \geq \texttt{opt} - \varepsilon$ *where* α_θ *is the translation that maximizes the overlap of* $\tilde{P}_1$ *and* $\tilde{P}_2$.

Proof. Define $\delta_\theta = 2D\sin\theta/\cos\psi$. Let α^* be the translation that maximizes the overlap of P_1 and P_2. Let α_θ be the translation that maximizes the overlap of $\tilde{P}_1$ and $\tilde{P}_2$.

For $j \in \{1, 2\}$, let S_j be the subset of P_j obtained by subtracting points at distance less than δ_θ from $\mathrm{bd}(P_j)$. We have

$$\mathrm{vol}(S_j) \ \geq \ \mathrm{vol}(P_j) - \sum_{f \in \mathrm{faces}(P_j)} V_{d-\dim(f)} \cdot \mathrm{vol}(f) \cdot \delta_\theta^{d-\dim(f)},$$

where $V_{d-\dim(f)}$ denotes the volume of a unit sphere of dimension $d - \dim(f)$. We take $V_0 = 0$ and $\mathrm{vol}(f) = 1$ if f is a vertex. We compute a threshold θ_ε in $O(n^{\lfloor d/2 \rfloor})$ time such that $\sum_{f \in \mathrm{faces}(P_j)} V_{d-\dim(f)} \cdot \mathrm{vol}(f) \cdot \delta_\theta^{d-\dim(f)} \leq \varepsilon/4$ whenever $\theta \leq \theta_\varepsilon$. In other words, $\mathrm{vol}(S_j) \geq \mathrm{vol}(P_j) - \varepsilon/4$ whenever $\theta \leq \theta_\varepsilon$. Therefore,

$$\mathrm{vol}((S_1 + \alpha^*) \cap S_2) \ \geq \ \mathrm{vol}((P_1 + \alpha^*) \cap P_2) - \varepsilon/2.$$

For $j \in \{1, 2\}$, let E_j denote the region obtained by adding to P_j points at distance δ_θ or less from the boundary of P_j. Arguing as in the previous paragraph, we can show that

$$\mathrm{vol}((E_1 + \alpha_\theta) \cap E_2) \ \leq \ \mathrm{vol}((P_1 + \alpha_\theta) \cap P_2) + \varepsilon/2.$$

By Lemma 10, $\mathrm{bd}(\tilde{P}_j)$ lies in $E_j \setminus S_j$. Therefore, $\mathrm{vol}((\tilde{P}_1 + \alpha^*) \cap \tilde{P}_2) \geq \mathrm{vol}((S_1 + \alpha^*) \cap S_2) \geq \mathrm{vol}((P_1 + \alpha^*) \cap P_2) - \varepsilon/2$. Since α_θ is the translation that maximizes the overlap of $\tilde{P}_1$ and $\tilde{P}_2$, we obtain

$$\begin{aligned} \mathrm{vol}((\tilde{P}_1 + \alpha_\theta) \cap \tilde{P}_2) \ &\geq \ \mathrm{vol}((\tilde{P}_1 + \alpha^*) \cap \tilde{P}_2) \\ &\geq \ \mathrm{vol}((P_1 + \alpha^*) \cap P_2) - \varepsilon/2. \end{aligned}$$

On the other hand, $\mathrm{vol}((\tilde{P}_1 + \alpha_\theta) \cap \tilde{P}_2) \leq \mathrm{vol}((E_1 + \alpha_\theta) \cap E_2) \leq \mathrm{vol}((P_1 + \alpha_\theta) \cap P_2) + \varepsilon/2$. Hence,

$$\begin{aligned} \mathrm{vol}((P_1 + \alpha_\theta) \cap P_2) \ &\geq \ \mathrm{vol}((\tilde{P}_1 + \alpha_\theta) \cap \tilde{P}_2) - \varepsilon/2 \\ &\geq \ \mathrm{vol}((P_1 + \alpha^*) \cap P_2) - \varepsilon. \end{aligned}$$

$\square$

A.3 Bounded incidence

In this section, we prove that $\tilde{P}_1$ and $\tilde{P}_2$ satisfy the bounded incidence condition with probability one. We prove the first part of the bounded incidence condition by showing that $\tilde{P}_j$ is a simple polytope, i.e., a k-face is contained in the intersection of exactly $d-k$ bounding hyperplanes. This implies that no more than 2^d faces of $\tilde{P}_j$ have a non-empty common intersection.

Lemma 12 *For $j \in \{1,2\}$, $\tilde{P}_j$ is a simple polytope with probability one.*

Proof. Let f be a proper face of $\tilde{P}_j$. Suppose that $\dim(f) > 0$. If there are more than $d - \dim(f)$ bounding hyperplanes containing f, the unit outward normals of any $d - \dim(f) + 1$ of them are not linearly independent. This happens with probability zero because these normals (at most d of them) are chosen randomly. Suppose that $\dim(f) = 0$, i.e., f is a vertex. There are d bounding hyperplanes with common intersection f. Consider the possibility that there is yet another bounding hyperplane $\tilde{h}$ passing through f. Since the anchor point a_h is picked at random, it is distinct from f with probability one. Then, since $\mathbf{n}_{\tilde{h}}$ is picked at random, the probability of $f \in \tilde{h}$ is zero. $\square$

In the rest of this section, Γ and $\widehat{\Gamma}$ are defined with respect to the perturbed polytopes $\tilde{P}_1$ and $\tilde{P}_2$ instead of the original polytopes P_1 and P_2.

We show in the next result that an element $\gamma_{f,g} \in \Gamma$ is formed generically with probability one in the sense that $\dim(\gamma_{f,g}) = \dim(f) + \dim(g)$. Recall that $\gamma_{f,g} = (-\mathrm{int}(f)) \oplus \mathrm{int}(g)$.

Lemma 13 *It holds with probability one that if $\dim(\gamma_{f,g}) < d$, then $\dim(\gamma_{f,g}) = \dim(f) + \dim(g)$.*

Proof. The lemma is trivial if $\dim(f)$ or $\dim(g)$ is zero. Assume that both $\dim(f)$ and $\dim(g)$ are positive. Since $\dim(\gamma_{f,g}) < d$, both $\dim(f)$ and $\dim(g)$ are less than d, implying that f and g are proper faces of P_1 and P_2, respectively. Let v be any vector parallel to $\mathrm{aff}(f)$. Let L be the $(d-1)$-dimensional linear subspace orthogonal to v. The intersection $L \cap \mathbb{S}^{d-1}$ is a unit $(d-2)$-sphere. Let $\tilde{h}$ be any bounding hyperplane of g. Since $\mathbf{n}_{\tilde{h}}$ is picked at random from a $(d-1)$-dimensional neighborhood on $\mathbb{S}^{d-1}$, the probability of $\mathbf{n}_{\tilde{h}} \in L \cap \mathbb{S}^{d-1}$ is zero. So v is not orthogonal to $\mathbf{n}_{\tilde{h}}$, meaning that v is not parallel to $\mathrm{aff}(g)$ with probability one. Conversely, any vector parallel to $\mathrm{aff}(g)$ is not parallel to $\mathrm{aff}(f)$ with probability one. It follows that $\dim(f \oplus g) = \dim(f) + \dim(g)$. Since $\gamma_{f,g} = (-\mathrm{int}(f)) \oplus \mathrm{int}(g)$, we have $\dim(\gamma_{f,g}) = \dim((-f) \oplus g) = \dim(f \oplus g) = \dim(f) + \dim(g)$. $\square$

We prove the second part of the bounded incidence condition by showing that no more than 2^{2d^2} hyperplanes in $\widehat{\Gamma}$ have a non-empty common intersection with probability one.

Lemma 14 *It holds with probability one that no more than 2^{2d^2} hyperplanes in $\widehat{\Gamma}$ have a non-empty common intersection.*

Proof. Let $\widehat{\mathcal{I}}_k$ be a subcollection of hyperplanes in $\widehat{\Gamma}$ that have a k-dimensional common intersection $\bigcap \widehat{\mathcal{I}}_k$, i.e., a k-flat. We prove below that $|\widehat{\mathcal{I}}_k| \geq 2^{2d(d-k)}$ with probability zero. Then, the lemma follows because $2^{2d(d-k)} \leq 2^{2d^2}$.

There must be $d-k$ hyperplanes in $\widehat{\mathcal{I}}_k$ whose common intersection is $\bigcap \widehat{\mathcal{I}}_k$. Let (f_1, g_1), $\ldots$, (f_{d-k}, g_{d-k}) be the pairs of faces that induce these hyperplanes. By Lemma 13, f_i and g_i are proper faces with probability one. Let $\tilde{L}_1$ denote the subset of hyperplanes in $\tilde{H}_1$ that contain f_i for some $i \in [1, d-k]$. By Lemma 12, it holds with probability one that there are at most d hyperplanes in $\tilde{H}_1$ containing each f_i. Thus, $|\tilde{L}_1| \leq d(d-k)$. There are fewer than $2^{d(d-k)}$ combinations of the hyperplanes in $\tilde{L}_1$, meaning that the union of the hyperplanes in

$\tilde{L}_1$ contains fewer than $2^{d(d-k)}$ faces of $\tilde{P}_1$. Similarly, let $\tilde{L}_2$ denote the subset of hyperplanes in $\tilde{H}_2$ that contain g_i for some $i \in [1, d-k]$ and the hyperplanes in $\tilde{L}_2$ contain fewer than $2^{d(d-k)}$ faces of $\tilde{P}_2$.

Therefore, in the event that $|\widehat{\mathcal{I}}_k| \geq 2^{2d(d-k)}$, some hyperplane in $\widehat{\mathcal{I}}_k$ must be induced by a pair of faces (f, g), where $f \in \text{faces}(P_1)$ and $g \in \text{faces}(P_2)$, such that f does not lie on any hyperplane in $\tilde{L}_1$ and g does not lie on any hyperplane in $\tilde{L}_2$. By Lemma 13, f and g are proper faces with probability one. Let $\tilde{L}$ denote the subset of hyperplanes in $\tilde{H}_1 \cup \tilde{H}_2$ that contain f or g. The important point is that the hyperplanes in $\tilde{L}$ are obtained by perturbations independent from the perturbations producing the hyperplanes in $\tilde{L}_1 \cup \tilde{L}_2$.

Without loss of generality, we translate space so that $\bigcap \widehat{\mathcal{I}}_k$ contains the origin.

We show in the following that the origin belongs to $\text{aff}(\gamma_{f,g})$ with probability zero. The origin belongs to $\text{aff}(\gamma_{f,g})$ if and only if $\text{aff}(f) \cap \text{aff}(g) \neq \emptyset$. By Lemma 12, $|\tilde{L}| = (d - \dim(f)) + (d - \dim(g))$. Since $\gamma_{f,g} \in \Gamma$, we have $\dim(\gamma_{f,g}) < d$ by definition and so $\dim(f) + \dim(g) = \dim(\gamma_{f,g}) < d$ by Lemma 13. Therefore, $|\tilde{L}| \geq d+1$. Since the normals and the anchor points of the hyperplanes in $\tilde{L}$ are picked at random, the intersection $\bigcap_{\tilde{h} \in \tilde{L}} \tilde{h}$ is empty with probability one. Notice that $\bigcap_{\tilde{h} \in \tilde{L}} \tilde{h}$ contains $\text{aff}(f) \cap \text{aff}(g)$. So $\text{aff}(f) \cap \text{aff}(g)$ is empty, which implies that the origin does not belong to $\text{aff}(\gamma_{f,g})$.

Recall that either $\widehat{\gamma}_{f,g} = \text{aff}(\gamma_{f,g})$ or $\widehat{\gamma}_{f,g}$ is a hyperplane containing $\text{aff}(\gamma_{f,g})$ picked at random. In the former case, as the origin belongs to $\bigcap \widehat{\mathcal{I}}_k$ but the origin does not belong to $\widehat{\gamma}_{f,g} = \text{aff}(\gamma_{f,g})$ with probability one, the probability of $\widehat{\gamma}_{f,g} \in \widehat{\mathcal{I}}_k$ is zero. In the latter case, as $\text{aff}(\gamma_{f,g})$ avoids the origin and $\widehat{\gamma}_{f,g}$ is picked at random, the probability of $\widehat{\gamma}_{f,g}$ containing the origin is zero, implying that the probability of $\widehat{\gamma}_{f,g} \in \widehat{\mathcal{I}}_k$ is zero. $\square$