

Approximate Convex Intersection Detection with Applications to Width and Minkowski Sums

Sunil Arya¹

Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
arya@cse.ust.hk

Guilherme D. da Fonseca²

Université Clermont Auvergne, LIMOS, and INRIA Sophia Antipolis, France
fonseca@isima.fr

David M. Mount³

Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD, USA
mount@cs.umd.edu

Abstract

Approximation problems involving a single convex body in \mathbb{R}^d have received a great deal of attention in the computational geometry community. In contrast, works involving multiple convex bodies are generally limited to dimensions $d \leq 3$ and/or do not consider approximation. In this paper, we consider approximations to two natural problems involving multiple convex bodies: detecting whether two polytopes intersect and computing their Minkowski sum. Given an approximation parameter $\varepsilon > 0$, we show how to independently preprocess two polytopes $A, B \subset \mathbb{R}^d$ into data structures of size $O(1/\varepsilon^{(d-1)/2})$ such that we can answer in polylogarithmic time whether A and B intersect approximately. More generally, we can answer this for the images of A and B under affine transformations. Next, we show how to ε -approximate the Minkowski sum of two given polytopes defined as the intersection of n halfspaces in $O(n \log(1/\varepsilon) + 1/\varepsilon^{(d-1)/2+\alpha})$ time, for any constant $\alpha > 0$. Finally, we present a surprising impact of these results to a well studied problem that considers a single convex body. We show how to ε -approximate the width of a set of n points in $O(n \log(1/\varepsilon) + 1/\varepsilon^{(d-1)/2+\alpha})$ time, for any constant $\alpha > 0$, a major improvement over the previous bound of roughly $O(n + 1/\varepsilon^{d-1})$ time.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Minkowski sum, convex intersection, width, approximation

Digital Object Identifier 10.4230/LIPIcs.ESA.2018.3

1 Introduction

Approximation problems involving a single convex body in d -dimensional space have received a great deal of attention in the computational geometry community [4, 9, 10, 11, 12, 18, 19, 45]. Recent results include near-optimal algorithms for approximating the convex hull of a set of points [9, 19], as well as an optimal data structure for answering approximate polytope

¹ Research supported by the Research Grants Council of Hong Kong, China under project number 16200014.

² Research supported by the European Research Council under ERC Grant Agreement number 339025 GUDHI (Algorithmic Foundations of Geometric Understanding in Higher Dimensions).

³ Research supported by NSF grant CCF-1618866.



membership queries [11]. In contrast, works involving multiple convex bodies are generally limited to dimensions $d \leq 3$ and/or do not consider approximation [2, 13, 29, 30, 44]. In this paper we present new approximation algorithms to natural problems that either involve multiple convex polytopes or result from such an analysis:

- Determining whether two convex polytopes A and B intersect
- Computing the Minkowski sum, $A \oplus B$, of two convex polytopes
- Computing the width of a convex polytope A (which results from an analysis of the Minkowski sum $A \oplus (-A)$)

Throughout we assume that the input polytopes reside in \mathbb{R}^d and are full-dimensional, where the dimension d is a fixed constant. Polytopes may be represented either as the convex hull of n points (*point representation*) or as the intersection of n halfspaces (*halfspace representation*). In either case, n denotes the *size* of the polytope.

1.1 Convex Intersection

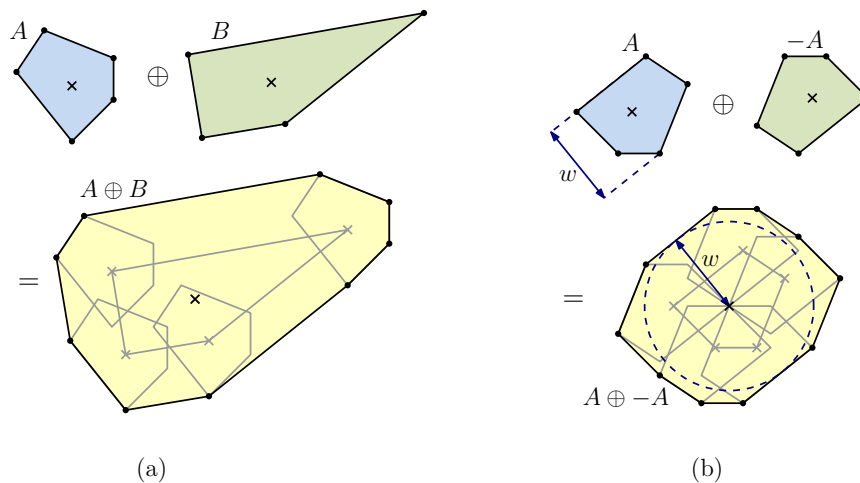
Detecting whether two geometric objects intersect and computing the region of intersection are fundamental problems in computational geometry. Geometric intersection problems arise naturally in a number of applications. Examples include geometric packing and covering, wire and component layout in VLSI, map overlay in geographic information systems, motion planning, and collision detection. Several surveys present the topics of collision detection and geometric intersection [33, 36, 37].

The special case of detecting the intersection of convex objects has received a lot of attention in computational geometry. The static version of the problem has been considered in \mathbb{R}^2 [39, 42] and \mathbb{R}^3 [20, 38]. The data structure version where each convex object is preprocessed independently has been considered in \mathbb{R}^2 [13, 21, 22, 25] and \mathbb{R}^3 [13, 22, 25, 26].

Recently, Barba and Langerman [13] considered the problem in higher dimension. They showed how to preprocess convex polytopes in \mathbb{R}^d so that given two such polytopes that have been subject to affine transformations, it can be determined whether they intersect each other in logarithmic time. However, the preprocessing time and storage grow as the combinatorial complexity of the polytope raised to the power $\lfloor d/2 \rfloor$. Since the combinatorial complexity of a polytope with n vertices can be as high as $\Theta(n^{\lfloor d/2 \rfloor})$, the storage upper bound is roughly $O(n^{d^2/4})$. This high complexity motivates the study of approximations to the problem.

We define approximation in a manner that is sensitive to direction. Consider any convex body K in \mathbb{R}^d and any $\varepsilon > 0$. Given a nonzero vector $v \in \mathbb{R}^d$, define $\Pi_v(K)$ to be the minimum slab defined by two hyperplanes that enclose K and are orthogonal to v . Define the *directional width* of K with respect to v , $\text{width}_v(K)$, to be the perpendicular distance between these hyperplanes. Let $\Pi_{v,\varepsilon}(K)$ be the central expansion of $\Pi_v(K)$ by a factor of $1 + \varepsilon$, and define K_ε to be the intersection of these expanded slabs over all unit vectors v . It can be shown that for any v , $\text{width}_v(K_\varepsilon) = (1 + \varepsilon) \text{width}_v(K)$. An ε -*approximation* of K is any set K' (which need not be convex) such that $K \subseteq K' \subseteq K_\varepsilon$. This defines an *outer* approximation. It is also possible to define an analogous notion of *inner* approximation in which each directional width is no smaller than $1 - \varepsilon$ times the true width. Our results can be extended to either type of approximation.

Given a discrete point set S in \mathbb{R}^d , an ε -*kernel* of S is any subset $Q \subseteq S$ such that $\text{conv}(Q)$ is an inner ε -approximation of $\text{conv}(S)$ [4]. It is well known that $O(1/\varepsilon^{(d-1)/2})$ points are sufficient and sometimes necessary in an ε -kernel. Kernels efficiently approximate the convex hull and as such have been used to obtain fast approximation algorithms to several problems such as diameter, minimum width, convex hull volume, minimum enclosing cylinder, minimum enclosing annulus, and minimum-width cylindrical shell [4, 5].



■ **Figure 1** Minkowski sum and its relationship to width.

In the ε -approximate version of convex intersection, we are given two convex bodies A and B and a parameter $\varepsilon > 0$. If $A \cap B \neq \emptyset$, then the answer is “yes.” If $A_\varepsilon \cap B_\varepsilon = \emptyset$, then the answer is “no.” Otherwise, either answer is acceptable. The ε -approximate polytope intersection problem is defined as follows. A collection of two or more convex polytopes in \mathbb{R}^d are individually preprocessed (with knowledge of ε). Given any two preprocessed polytopes, A and B , the query determines whether A and B intersect approximately. In general, the query algorithm can be applied to any affine transformation of the preprocessed polytopes.

► **Theorem 1.** *Given a parameter $\varepsilon > 0$ and two polytopes $A, B \subset \mathbb{R}^d$ each of size n (given either using a point or halfspace representation), we can independently preprocess each polytope into a data structure in order to answer ε -approximate polytope intersection queries with query time $O(\text{polylog } \frac{1}{\varepsilon})$, storage $O(1/\varepsilon^{(d-1)/2})$, and preprocessing time $O(n \log \frac{1}{\varepsilon} + 1/\varepsilon^{(d-1)/2+\alpha})$, where α is an arbitrarily small positive constant.*

The space is nearly optimal because there is a lower bound of $\Omega(1/\varepsilon^{(d-1)/2})$ on the worst-case bit complexity of representing an ε -approximation of a polytope [11].

1.2 Minkowski Sum

Given two convex bodies $A, B \subset \mathbb{R}^d$, the *Minkowski sum* $A \oplus B$ is defined as $\{p + q : p \in A, q \in B\}$ (see Figure 1(a)). Minkowski sums have found numerous applications in motion planning [7, 31], computer-aided design [44], computational biology [40], satellite layout [15], and image processing [35]. Minkowski sums have been also well studied in the context of discrete and computational geometry [1, 3, 29, 32, 43].

It is well known that in dimension $d \geq 3$, the number of vertices in the Minkowski sum of two polytopes can grow as rapidly as the product of the number of vertices in the two polytopes [7]. This has led to the study of algorithms to compute approximations to Minkowski sums in \mathbb{R}^3 [2, 30, 44]. In this paper, we show how to approximate the Minkowski sum of two convex polytopes in \mathbb{R}^d in near-optimal time.

► **Theorem 2.** *Given a parameter $\varepsilon > 0$ and two polytopes $A, B \subset \mathbb{R}^d$ each of size n (given either using a point or halfspace representation), it is possible to construct an ε -approximation of $A \oplus B$ of size $O(1/\varepsilon^{(d-1)/2})$ in $O(n \log \frac{1}{\varepsilon} + 1/\varepsilon^{(d-1)/2+\alpha})$ time, where α is an arbitrarily small positive constant.*

The output representation can be either point-based or halfspace-based.

1.3 Width

Define the directional width of a set S of n points to be the directional width of $\text{conv}(S)$. The *width* of S is the *minimum* over all directional widths. The *maximum* over all directional widths is equal to the diameter of S . Both problems can be approximated using the ε -kernel of S . After successive improvements [4, 6, 8, 14, 18], algorithms to compute ε -kernels and to ε -approximate the diameter in roughly $O(n+1/\varepsilon^{d/2})$ time have been independently discovered by Chan [19] and the authors [9]. Somewhat surprisingly, these works offer no improvement to the running time to approximate the width [4, 17, 18, 28, 45], which Chan [19] posed as an open problem. The fastest known algorithms date from over a decade ago and take roughly $O(n+1/\varepsilon^{d-1})$ time [17, 18].

Agarwal *et al.* [2] showed that the width of a convex body K is equal to the minimum distance from the origin to the boundary of the convex body $K \oplus (-K)$ (see Figure 1(b)). Using Theorem 2, we can approximate the width by computing an ε -approximation of $K \oplus (-K)$ represented as the intersection of halfspaces and then determining the closest point to the origin among all bounding hyperplanes. The following presents this result.

► **Theorem 3.** *Given a set S of n points in \mathbb{R}^d and an approximation parameter $\varepsilon > 0$, it is possible to compute an ε -approximation to the width of S in $O(n \log \frac{1}{\varepsilon} + 1/\varepsilon^{(d-1)/2+\alpha})$ time, where α is an arbitrarily small positive constant.*

1.4 Techniques

Our algorithms and data structure are based on a data structure defined by a hierarchy of Macbeath regions [9, 11], which answers approximate directional width queries in polylogarithmic time. First, we show how to use this data structure as a black box to answer approximate polytope intersection queries by transforming the problem to a dual setting and performing a multidimensional convex minimization. Next, we show how to use approximate polytope intersection queries to compute ε -approximations of the Minkowski sum. The approximation to the width follows directly.

Since we only access the input polytopes through a data structure for approximate directional width queries, our results apply in much more general settings. For example, we could answer in polylogarithmic time whether the Minkowski sum of two polytopes (preprocessed independently) approximately intersects a third polytope. Our techniques are also amenable to other polytope operations such as intersection and convex hull of the union, as long as the model of approximation is defined accordingly.

The preprocessing time of the approximate directional width data structure we use is $O(n \log \frac{1}{\varepsilon} + 1/\varepsilon^{(d-1)/2+\alpha})$, for arbitrarily small $\alpha > 0$. If this preprocessing time is reduced in the future, the complexity of our algorithms becomes equal to the preprocessing time plus $O((1/\varepsilon^{(d-1)/2}) \text{polylog } \frac{1}{\varepsilon})$.

2 Preliminaries

In this section we present a number of results, which will be used throughout the paper. The first provides three basic properties of Minkowski sums. The proof can be found in standard sources on Minkowski sums (see, e.g., [41]).

- **Lemma 4.** *Let $A, B \subset \mathbb{R}^d$ be two (possibly infinite) sets of points. Then:*
- (a) $A \cap B \neq \emptyset$ if and only if $O \in A \oplus (-B)$, where O is the origin.
 - (b) $\text{conv}(A \oplus B) = \text{conv}(A) \oplus \text{conv}(B)$.
 - (c) For all nonzero vectors v , $\text{width}_v(A \oplus B) = \text{width}_v(A) + \text{width}_v(B)$.

Next, we recall a recent result of ours on answering directional width queries approximately [9], which we will use as a black box later in this paper. Given a set S of n points in a constant dimension d and an approximation parameter $\varepsilon > 0$, the answer to the *approximate directional width query* for a nonzero query vector v consists of a pair of points $p, q \in S$ such that $\text{width}_v(\{p, q\}) \geq (1 - \varepsilon) \text{width}_v(S)$.

► **Lemma 5.** *Given a set S of n points in \mathbb{R}^d and an approximation parameter $\varepsilon > 0$, there is a data structure that can answer ε -approximate directional width queries with query time $O(\log^2 \frac{1}{\varepsilon})$, space $O(1/\varepsilon^{(d-1)/2})$, and preprocessing time $O(n \log \frac{1}{\varepsilon} + 1/\varepsilon^{(d-1)/2+\alpha})$.*

2.1 Fattening

Existing algorithms and data structures for convex approximation often assume that the bodies have been fattened through an appropriate affine transformation. In the context of multiple bodies, this is complicated by the fact that different fattening transformations may be needed for the two bodies or their Minkowski sum. In this section we explore this issue.

Consider a convex body K in d -dimensional space \mathbb{R}^d . Given a parameter $0 < \gamma \leq 1$, we say that K is γ -fat if there exist concentric Euclidean balls B and B' , such that $B \subseteq K \subseteq B'$, and $\text{radius}(B)/\text{radius}(B') \geq \gamma$. We say that K is *fat* if it is γ -fat for a constant γ (possibly depending on d , but not on ε or K). For a centrally symmetric convex body C , the body obtained by scaling C about its center by a factor of λ is called the λ -expansion of C .

Let K be a convex body. We say that a convex body C is a λ -sandwiching body for K if C is centrally symmetric and $C \subseteq K \subseteq C'$, where C' is a λ -expansion of C . John [34] proved tight bounds for the constant λ of a λ -sandwiching ellipsoid. This ellipsoid is referred to as the *John ellipsoid*.

► **Lemma 6.** *For every convex body K in \mathbb{R}^d , there exists a d -sandwiching ellipsoid. Furthermore, if K is centrally symmetric, there exists a \sqrt{d} -sandwiching ellipsoid.*

It is an immediate consequence of this lemma that for any convex body K there exists an affine transformation T such that $T(K)$ is $(1/d)$ -fat. Any affine transformation that maps the John ellipsoid into a Euclidean ball will do. The following lemma generalizes this to hyperrectangles (see also Barequet and Har-Peled [14]).

► **Lemma 7.** *For every convex body K in \mathbb{R}^d , there exists a $(d^{3/2})$ -sandwiching hyperrectangle.*

Proof. Let E denote the d -sandwiching ellipsoid for K , described in Lemma 6. By elementary geometry, there exists a \sqrt{d} -sandwiching hyperrectangle R for E . We claim that R is a $(d^{3/2})$ -sandwiching hyperrectangle for K . To prove this claim, observe that $R \subseteq E \subseteq R'$ and $E \subseteq K \subseteq E'$, where R' is the \sqrt{d} -expansion of R and E' is the d -expansion of E . Letting R'' denote the d -expansion of R' , it is easy to see that $E' \subseteq R''$. It follows that $R \subseteq E \subseteq K \subseteq E' \subseteq R''$. Since R'' is the d -expansion of R' and R' is the \sqrt{d} -expansion of R , it follows that R'' is the $(d^{3/2})$ -expansion of R . This completes the proof. ◀

Next, let us consider fattening in the context of multiple bodies. The next two lemmas follow from elementary geometry and properties of Minkowski sums.

► **Lemma 8.** *Let C_1 and C_2 be λ -sandwiching bodies for K_1 and K_2 , respectively. Then $C_1 \oplus C_2$ is a λ -sandwiching body for $K_1 \oplus K_2$.*

► **Lemma 9.** *Let K be a convex body. Given a λ -sandwiching polytope for K of constant complexity, we can compute a γ -fattening affine transformation T for K in constant time, where $\gamma = 1/(\lambda\sqrt{d})$.*

We conclude by showing that we can maintain a small amount of auxiliary information for any collection of convex bodies in order to determine the fattening transformation for the Minkowski sum of any two members of this library. We refer to the data structure for approximate directional width queries from Lemma 5 together with the additional information to determine the fattening transformation as the *augmented data structure* for approximate directional width queries.

► **Lemma 10.** *Consider any finite collection of convex polytopes in \mathbb{R}^d , and let $\gamma = 1/d^2$. It is possible to store information of constant size with each polytope such that in constant time we can compute a γ -fattening affine transformation for the Minkowski sum of any two polytopes from the collection. This information can be computed in time proportional to the size of the input polytope.*

Proof. At preprocessing time, we store the λ -sandwiching hyperrectangles R_i for each K_i , where $\lambda = d^{3/2}$. By Lemma 7, such hyperrectangles exist and they can be computed in time proportional to the size of the input polytope [23].

Suppose we want to compute a γ -fattening affine transformation for $K'_i \oplus K'_j$, where K'_i and K'_j are the result of applying (possibly different) affine transformations to K_i and K_j , respectively. Let C'_i and C'_j be the polytopes of constant complexity obtained by applying the corresponding affine transformations to R_i and R_j , respectively. Clearly, C'_i and C'_j are λ -sandwiching polytopes for K'_i and K'_j , respectively. Thus, by Lemma 8, $C'_i \oplus C'_j$ is a λ -sandwiching polytope for $K'_i \oplus K'_j$. Note that this polytope has constant complexity and can be computed in constant time. Applying Lemma 9, we can use this polytope to compute a γ -fattening affine transformation for $K'_i \oplus K'_j$ in constant time, where $\gamma = 1/(\lambda\sqrt{d}) = 1/d^2$. ◀

The previous lemma holds more generally even when each of the polytopes are subject to any non-singular affine transformation and to the Minkowski sum of a constant number of polytopes.

2.2 Projective Duality and Width

Our algorithm for approximating the directional width of a point set is based on a *projective dual transformation*, which maps points into hyperplanes and vice versa. Each primal point $p = (p_1, \dots, p_d) \in S$ is mapped to the dual hyperplane $p^* : x_d = p_1x_1 + \dots + p_{d-1}x_{d-1} - p_d$. Each primal hyperplane is mapped to a dual point in the same manner. This dual transformation has several well-known properties [24]. For example, the points in the lower convex hull of S map to the hyperplanes in the upper envelope.

Let H be a set of n hyperplanes in \mathbb{R}^d . Given a point $r \in \mathbb{R}^{d-1}$, the *thickness* of H at r , denoted $\text{thick}_r(H)$ is defined as follows. Given $r \in \mathbb{R}^{d-1}$ and $t \in \mathbb{R}$, let (r, t) denote the point in \mathbb{R}^d resulting by concatenating r and t . For the sake of illustration, we think of the d -th coordinate axis as being the vertical axis. Let $r' = (r, t_1)$ and $r'' = (r, t_2)$. We define $\text{thick}_r(H)$ as the maximum difference $t_2 - t_1$ for points r', r'' in the hyperplanes in H . In other words, the thickness is the vertical distance between the intersection of the vertical line defined by r with the upper and lower envelopes of H . The following relates width and thickness.

► **Lemma 11.** *Consider two points $p, q \in \mathbb{R}^d$ and a vector $v = (v_1, \dots, v_{d-1}, -1)$. Let p^*, q^* denote the dual hyperplanes and $v_{1,d-1} = (v_1, \dots, v_{d-1})$. We have*

$$\text{thick}_{v_{1,d-1}}(\{p^*, q^*\}) = \|v\| \text{width}_v(\{p, q\}).$$

Proof. Given vectors u and v , let $u \cdot v$ denote the standard inner product. Assume without loss of generality that $p \cdot v \geq q \cdot v$. Clearly, v is nonzero, so $\text{width}_v(\{p, q\}) = (p \cdot v - q \cdot v) / \|v\|$. Let $p = (p_1, \dots, p_d)$ and $q = (q_1, \dots, q_d)$. The dual hyperplanes are

$$p^* : x_d = p_1x_1 + \dots + p_{d-1}x_{d-1} - p_d \quad \text{and} \quad q^* : x_d = q_1x_1 + \dots + q_{d-1}x_{d-1} - q_d.$$

If we set $x_1, \dots, x_{d-1} = v_{1,d-1}$ we have $t_2 = (p_1, \dots, p_{d-1}) \cdot v_{1,d-1} - p_d$ and $t_1 = (q_1, \dots, q_{d-1}) \cdot v_{1,d-1} - q_d$. Therefore

$$\begin{aligned} \text{thick}_{v_{1,d-1}}(H) &= t_2 - t_1 \\ &= (p_1, \dots, p_{d-1}) \cdot v_{1,d-1} - p_d - ((q_1, \dots, q_{d-1}) \cdot v_{1,d-1} - q_d) \\ &= p \cdot v - q \cdot v \\ &= \|v\| \text{width}_v(\{p, q\}). \end{aligned} \quad \blacktriangleleft$$

3 Approximate Convex Intersection

In this section, we will prove Theorem 1 for the case when the input polytopes are represented by points. Assume that we are given two polytopes A and B in the point representation. The objective is to preprocess A and B individually such that we can efficiently answer approximate intersection queries for A and B (or more generally for affine transformations of A and B).

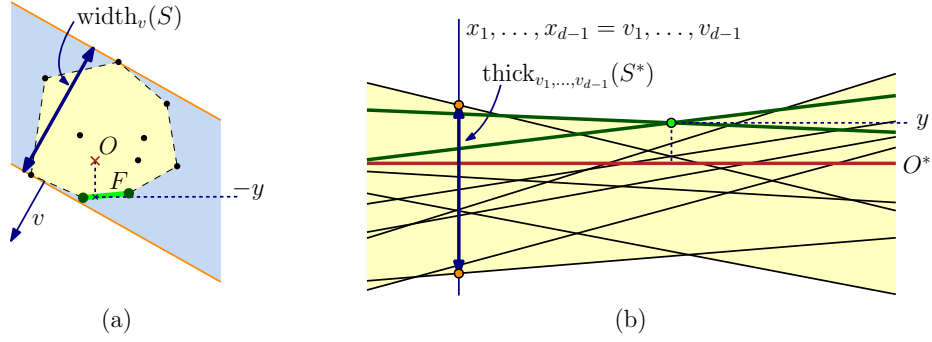
Given a convex body K , $\varepsilon > 0$, and a point p , an ε -approximate polytope membership query is defined as follows. If $p \in K$, the answer is “yes,” if $p \notin K_\varepsilon$, the answer is “no,” and otherwise, either answer is acceptable. Our strategy to answer approximate intersection queries is based on reducing them to approximate polytope membership queries. This reduction is presented in the following lemma, which is a straightforward generalization of Lemma 4(a) to an approximate context. The proof follows from standard algebraic properties of Minkowski sums and the observation that K_ε can be expressed as $K \oplus \frac{\varepsilon}{2}(K \oplus -K)$, and is omitted from this version.

► **Lemma 12.** *Let $A, B \subset \mathbb{R}^d$ be two polytopes and $\varepsilon > 0$. Determining the ε -approximate intersection of A and B is equivalent to determining the ε -approximate membership of $O \in A \oplus (-B)$.*

The previous lemma relates approximate polytope intersection with an approximate membership of the origin in a polytope (Figure 2(a)). Determining whether the origin lies within the convex hull of a set of points S is a classic problem in computational geometry, which can be solved by linear programming. However, we are interested in a faster approximate solution that does not compute S explicitly. We cannot afford to preprocess an approximate polytope membership data structure for $A \oplus (-B)$ for each pair A and B , since the number of such pairs is quadratic in the number of input polytopes. Instead, we preprocess each input polytope individually, and we show next how to efficiently answer approximate polytope membership queries for $A \oplus (-B)$ by using augmented data structures for approximate directional width queries for A and B as black boxes.

► **Lemma 13.** *Given augmented data structures for answering ε -approximate directional width queries for polytopes A and B , we can answer ε -approximate membership queries for $A \oplus (-B)$ using $O(\text{polylog } \frac{1}{\varepsilon})$ queries to these data structures.*

Proof. Without loss of generality, we may translate space so that the query point coincides with the origin O . Let $K = A \oplus (-B)$, and let S be K 's vertex set. (Note that K and S are not explicitly computed.)



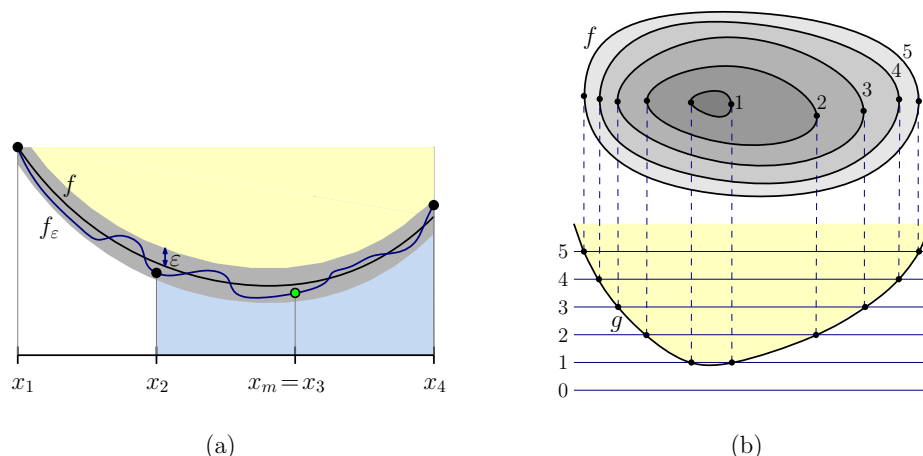
■ **Figure 2** (a) Primal problem of determining if $O \in \text{conv}(S)$. (b) Dual problem of determining if the horizontal hyperplane O^* is between the upper and lower envelopes.

The problem of determining whether $O \in K$ is invariant to scaling and rotation about the origin. It will be helpful to perform some affine transformations that will guarantee certain properties for K . First, we apply Lemma 10 to fatten K and then apply a uniform scaling about the origin so that K 's diameter is $\Theta(1)$. By fatness, K has a λ -sandwiching ball of radius $r = \Theta(1)$. If the origin either lies within the inner ball or outside the outer ball, then the answer is trivial. Otherwise, let $\Delta = 2\lambda r$ be the diameter of the outer ball. We may apply a rotation about the origin so that the center of this ball lies on the positive x_d axis at a point $(0, \dots, 0, \beta)$. Again, this scaling and rotation can be computed in constant time using the augmented information. It follows that the coordinates of the points of S have absolute values at most $\Delta = \Theta(1)$.

In summary, there exists an affine transformation computable in constant time such that after applying this transformation, the query point lies at the origin, $K = \text{conv}(S)$ is sandwiched between two concentric balls of constant radii centered at $c = (0, \dots, 0, \beta)$, where $0 < \beta \leq \Delta = O(1)$, and K 's vertex set S is contained within $[-\Delta, \Delta]^d$. It is an immediate consequence that $\text{width}_v(K) = \Theta(1)$ for all directions v , and hence it suffices to answer the membership query to an absolute error of $\Theta(\varepsilon)$.

Lemma 4(c) implies that we can answer ε -approximate width queries for K as the sum of two ε -approximate width queries to A and B . Therefore, our goal is to determine approximately if $O \in K$ using only approximate width queries to A and B . In order to do this, we look at the projective dual problem in which each point $p = (p_1, \dots, p_d) \in S$ is mapped to the hyperplane $p^* : x_d = p_1x_1 + \dots + p_{d-1}x_{d-1} - p_d$. Let S^* denote the corresponding set of hyperplanes. The primal problem $O \in K$ is equivalent to the dual problem of determining whether the horizontal hyperplane $O^* : x_d = 0$ is sandwiched between the upper and lower envelopes of S^* (Figure 2(b)). Since the point c lies vertically above the origin and within K 's interior, it follows that O^* cannot intersect the lower envelope. Therefore, it suffices to test whether O^* intersects the upper envelope.

The dual problem can be solved exactly by computing the minimum value y of the x_d -coordinate in the upper envelope and testing whether $y > 0$. In the primal, the value of y corresponds to the negated x_d -coordinate of the intersection of a facet F of the lower convex hull of K and a vertical line passing through the origin (see Figure 2). Let F 's supporting hyperplane be denoted by $x_d = w_1x_1 + \dots + w_{d-1}x_{d-1} - w_d$. Since K is sandwiched between two concentric balls of constant radii whose common center lies on this vertical line, it follows from simple geometry that this supporting hyperplane cannot be very steep. In particular, there exists $\alpha = O(1)$ such that $w_i \in [-\alpha, \alpha]$, for $i = 1, \dots, d-1$. In the dual, this means that the minimum value y is attained at a point whose first $d-1$ coordinates all lie within



■ **Figure 3** (a) One-dimensional convex minimization. (b) Higher-dimensional convex minimization.

$[-\alpha, \alpha]$. In approximating y , we will apply directional width queries only for directional vectors $v = (v_1, \dots, v_d)$ whose first $d-1$ coordinates lie within $[-\alpha, \alpha]$ and $v_d = -1$. Thus, $\|v\| = O(1)$.

By Lemma 11, the duals of two points $p, q \in S$ returned by an exact directional width query $\text{width}_v(K)$ in the primal for a vector $v = (v_1, \dots, v_{d-1}, -1)$ correspond to the two dual hyperplanes in the upper and lower envelopes of S^* that intersect the vertical line $x_i = v_i$ for $i = 1, \dots, d-1$. Since queries are only applied to directions v where $\|v\| = O(1)$ and since $\text{width}_v(K) = \Theta(1)$ for all directions v , it follows from Lemma 11 that a relative error of ε in the directional width implies an absolute error of $O(\varepsilon)$ in the corresponding thickness. We can think of the upper envelope of S^* as defining the graph of a convex function over the domain $[-\alpha, \alpha]^{d-1}$. Since $S \subset [-\Delta, \Delta]^d$, the slopes of the hyperplanes in S^* are similarly bounded, and therefore this function has bounded slope. It follows that, for an appropriate $\varepsilon' = \Theta(\varepsilon)$, we can compute this function to an absolute error of ε at any (v_1, \dots, v_{d-1}) by performing an (ε') -approximate directional width query on K for $v = (v_1, \dots, v_{d-1}, -1)$. To complete the proof, it suffices to show that with $O(\text{polylog} \frac{1}{\varepsilon})$ such queries, it is possible to compute an absolute ε -approximation to y . We do this in the next section. ◀

3.1 Convex Minimization

The following lemma shows how to use binary search to solve a one-dimensional convex minimization problem approximately (see Figure 3(a)).

► **Lemma 14.** *Let $a, b \in \mathbb{R}$ and $\varepsilon \in \mathbb{R}^+$ be real parameters. Let $f : [a, b] \rightarrow \mathbb{R}$ be a convex function with bounded slope and $f_\varepsilon : [a, b] \rightarrow \mathbb{R}$ be a function with $|f(x) - f_\varepsilon(x)| \leq \varepsilon$ for all $x \in [a, b]$. Let $x^* \in [a, b]$ be the value of x that minimizes $f(x)$. It is possible to determine a value x' with $f(x') - f(x^*) = O(\varepsilon)$ after $O(\log((b-a)/\varepsilon))$ evaluations of $f_\varepsilon(\cdot)$ and no evaluation of $f(\cdot)$.*

Proof. First, we present the recursive algorithm used to determine the value x' . If $b-a < \varepsilon$, then since the function has bounded slope, we simply return $x' = a$, as a valid answer.

Otherwise, we start by trisecting the interval $[a, b]$ and evaluate $f_\varepsilon(x)$ at the four endpoints x_1, x_2, x_3, x_4 of the subintervals (see Figure 3(a)). Let m denote the value i that minimizes $f_\varepsilon(x_i)$, breaking ties arbitrarily. To simplify the boundary cases, let $x_0 = a$ and $x_5 = b$.

3:10 Approximate Convex Intersection Detection with Applications

We then invoke our algorithm recursively on the interval $[x_{m-1}, x_{m+1}]$ and store the value returned as x'' . We return the value x among the two values x_m, x'' that minimizes $f_\varepsilon(x)$.

Since the length of the interval reduces by at least one third at each iteration, the number of recursive calls and therefore evaluations of $f_\varepsilon(\cdot)$ is $O(\log((b-a)/\varepsilon))$. Next, we show that $f(x') - f(x^*) = O(\varepsilon)$. By the convexity of f we have

$$f(x) \geq f(x_{m+1}) + 3(x - x_{m+1})(f(x_{m+1}) - f(x_m))/(b - a), \text{ for } x \geq x_{m+1}.$$

Using that $|f(x) - f_\varepsilon(x)| \leq \varepsilon$, we have

$$f(x) \geq f_\varepsilon(x_{m+1}) - \varepsilon + 3(x - x_{m+1})(f_\varepsilon(x_{m+1}) - f_\varepsilon(x_m) - 2\varepsilon)/(b - a), \text{ for } x \geq x_{m+1}.$$

Since $f_\varepsilon(x_m) \leq f_\varepsilon(x_{m+1})$, we have

$$f(x) \geq f_\varepsilon(x_m) - \varepsilon - 6\varepsilon(x - x_{m+1})/(b - a), \text{ for } x \geq x_{m+1}.$$

For x inside the interval $[a, b]$ we have $|x - x_{m+1}| \leq b - a$, and therefore

$$f(x) \geq f_\varepsilon(x_m) - 7\varepsilon, \text{ for } x_{m+1} \leq x \leq b.$$

The same argument is used to bound the case of $a \leq x \leq x_{m-1}$, obtaining

$$f(x) \geq f_\varepsilon(x_m) - 7\varepsilon, \text{ for } x \notin [x_{m-1}, x_{m+1}].$$

Either the minimum of $f(x)$ is inside the interval $[x_{m-1}, x_{m+1}]$ or not. If it is not, then the previous inequality shows that $f_\varepsilon(x_m)$ provides a good approximation, regardless of the value returned in the recursive call. If the minimum is inside the interval $[x_{m-1}, x_{m+1}]$, then the recursive call will provide a value result by an inductive argument. ◀

We are now ready to extend the result to arbitrary dimensions.

► **Lemma 15.** *Let $a, b \in \mathbb{R}$ and $\varepsilon \in \mathbb{R}^+$ be real parameters. Let $f : [a, b]^d \rightarrow \mathbb{R}$ for a constant dimension d be a convex function with bounded slope and $f_\varepsilon : [a, b]^d \rightarrow \mathbb{R}$ be a function with $|f(x) - f_\varepsilon(x)| \leq \varepsilon$ for all $x \in [a, b]^d$. Let $x^* \in [a, b]^d$ be the value of x that minimizes $f(x)$. It is possible to determine a value x' with $f(x') - f(x^*) = O(\varepsilon)$ after $O(\log^d((b-a)/\varepsilon))$ evaluations of $f_\varepsilon(\cdot)$ and no evaluation of $f(\cdot)$.*

Proof. The minimum $f(x^*)$ can be written as

$$f(x^*) = \min_{x \in [a, b]^d} f(x) = \min_{x_1 \in [a, b]} \min_{\tilde{x} \in [a, b]^{d-1}} f(x_1, \tilde{x}).$$

Note that if $f(x)$ is a convex function with bounded slope, then so is the function $g : [a, b] \rightarrow \mathbb{R}$ (see Figure 3(b)) defined as

$$g(x_1) = \min_{\tilde{x} \in [a, b]^{d-1}} f(x_1, \tilde{x}).$$

The proof is based on induction on the dimension d . Since d is a constant, the number of induction steps is also a constant. The base case of $d = 1$ follows from Lemma 14. By the induction hypothesis, we can solve the $(d-1)$ -dimensional instance to obtain a function $g'(x_1)$ such that

$$|g(x_1) - g'(x_1)| = O(\varepsilon).$$

Using Lemma 14 for the function $g'(\cdot)$, we obtain a value x' with $f(x') - f(x^*) = O(\varepsilon)$.

For the number of function evaluations $t(d)$ for a given dimension d we have

$$t(1) = O(\log((b - a)/\varepsilon)) \text{ and}$$

$$t(k) = t(1) \cdot t(k - 1).$$

The recurrence easily solves to the desired

$$t(d) = O(\log^d((b - a)/\varepsilon)). \quad \blacktriangleleft$$

By applying Lemma 15 to the dual problem defined in the proof of Lemma 13 (where f is the graph of the upper envelope of S^* and $[a, b] = [-\alpha, \alpha]$) with the augmented data structure from Lemma 5, we obtain Theorem 1 for the case when the input polytopes are represented by points. We will consider the case when the input polytopes are represented by halfspaces at the end of the next section.

4 Minkowski Sum Approximation

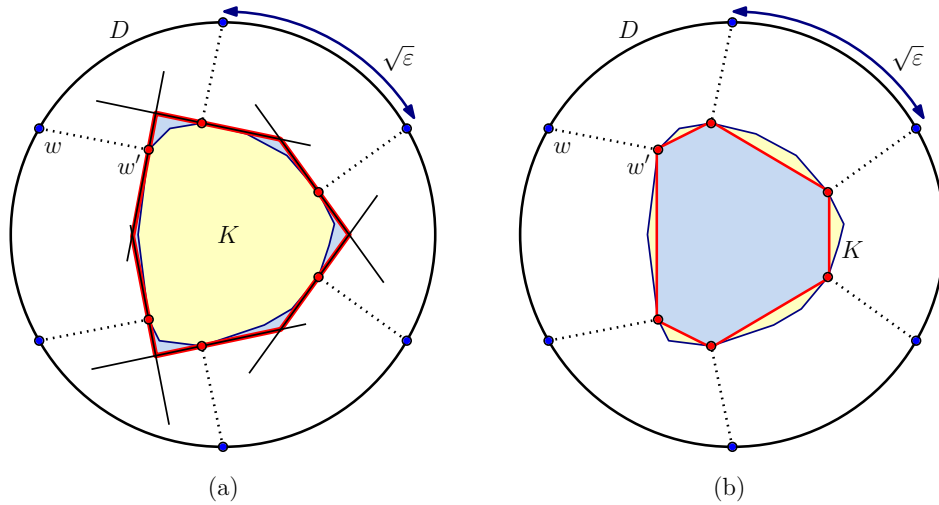
In this section, we will prove Theorems 2 and 3, as well as Theorem 1 for the case when the input polytopes are represented by halfspaces. Assume that we are given two polytopes A and B in the point representation, and we have computed the augmented approximate directional width data structures from Lemma 5 for each polytope. The objective is to obtain an ε -approximation of the Minkowski sum $A \oplus B$ of size $O(1/\varepsilon^{(d-1)/2})$ using these data structures. Our approach is to fatten $A \oplus B$ using Lemma 10 and then apply Dudley's construction [27] in order to obtain an approximation with $O(1/\varepsilon^{(d-1)/2})$ halfspaces. For completeness, we start by describing Dudley's algorithm.

Let $K \subset [-1, 1]^d$ be a fat polytope of constant diameter. Dudley's algorithm obtains an ε -approximation represented by halfspaces as follows. Let D be a ball of radius $2\sqrt{d}$ centered at the origin. (Note that $K \subset D$.) Place a set W of $\Theta(1/\varepsilon^{(d-1)/2})$ points on the surface of D such that every point on the surface of D is within distance $O(\sqrt{\varepsilon})$ of some point in W . For each point $w \in W$, let w' be its nearest point on the boundary of K . We call these points *samples*. For each sample point w' , take the supporting halfspace passing through w' that is orthogonal to the vector from w' to w . The approximation is defined as the intersection of these halfspaces (see Figure 4(a)).

Bronshteyn and Ivanov [16] presented a similar construction. Instead of approximating K by halfspaces, Bronshteyn and Ivanov's construction approximates K as the convex hull of the aforementioned set of samples⁴ (see Figure 4(b)). In both constructions it is possible to tune the constant factors so that closest point queries need only be computed to within an absolute error of $\Theta(\varepsilon)$.

An approximate closest point query between a polytope K and a point p within constant distance from K can be reduced to computing an ε -approximation to the smallest radius ball centered at p that intersects K . This can be solved through binary search on the radius of this ball, where each probe involves determining whether K intersects a ball of some radius centered at p . Notice that the data structure for approximate polytope intersection from Section 3 only accesses the bodies through approximate directional width queries, besides the initial fattening transformation. By Lemma 4(c), given two preprocessed bodies A and

⁴ Dudley's construction yields an outer approximation and Bronshteyn and Ivanov's yields inner approximation, but it is possible to convert both to the other type through standard techniques. For details, see Lemma 2.8 of the full version of [9].



■ **Figure 4** (a) Dudley's and (b) Bronshteyn and Ivanov's polytope approximations.

B , we can answer directional width queries on $A \oplus B$ through directional width queries on A and B individually. (In the case of a ball, no data structure is required.) Therefore, we can test intersection with a Minkowski sum $A \oplus B$, as long as we have augmented approximate directional width data structures for both A and B .

In order to establish Theorem 2 for the case when the input polytopes are represented by points, we apply the aforementioned binary search to simulate Dudley's construction. Each sample is obtained after $O(\log \frac{1}{\epsilon})$ ϵ -approximate polytope intersection queries. The total running time is dominated by the preprocessing time of Lemma 5. Note that the output polytope may be represented by either points or halfspaces according to whether we use Dudley's or Bronshteyn and Ivanov's algorithm. To show that the input polytopes may be represented by halfspaces, we show how to efficiently convert between the two representations.

► **Lemma 16.** *Given an approximation parameter $\epsilon > 0$ and a polytope $K \subset \mathbb{R}^d$ of size n (given either using a point or halfspace representation), we can obtain an ϵ -approximation of size $O(1/\epsilon^{(d-1)/2})$ (in either representation, independent of the input representation) in $O(n \log \frac{1}{\epsilon} + 1/\epsilon^{(d-1)/2+\alpha})$ time, where $\alpha > 0$ is an arbitrarily small constant.*

Proof. The case when the input is represented by points is a trivial case of Theorem 2, where $B = \{O\}$. For the alternative case, it suffices to obtain an ϵ -approximation of the polar polytope after fattening. (For details see Lemma 2.9 of the full version of [9].) ◀

We remind the reader that Agarwal *et al.* [2] showed that the width of a convex body K is equal to the minimum distance from the origin to the boundary of the convex body $K \oplus (-K)$. To obtain Theorem 3, we compute Dudley's approximation of $K \oplus (-K)$ and then we determine the closest point to the origin among the $O(1/\epsilon^{(d-1)/2})$ bounding hyperplanes of the approximation.

— **References** —

- 1 P. K. Agarwal, E. Flato, and D. Halperin. Polygon decomposition for efficient construction of Minkowski sums. *Comput. Geom. Theory Appl.*, 21(1):39–61, 2002.
- 2 P. K. Agarwal, L. J. Guibas, S. Har-Peled, A. Rabinovitch, and M. Sharir. Penetration depth of two convex polytopes in 3D. *Nordic J. of Computing*, 7(3):227–240, 2000.

- 3 P. K. Agarwal, S. Har-Peled, H. Kaplan, and M. Sharir. Union of random Minkowski sums and network vulnerability analysis. *Discrete Comput. Geom.*, 52(3):551–582, 2014.
- 4 P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Approximating extent measures of points. *J. Assoc. Comput. Mach.*, 51:606–635, 2004.
- 5 P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Geometric approximation via core-sets. In J. E. Goodman, J. Pach, and E. Welzl, editors, *Combinatorial and Computational Geometry*. MSRI Publications, 2005.
- 6 P. K. Agarwal, J. Matoušek, and S. Suri. Farthest neighbors, maximum spanning trees and related problems in higher dimensions. *Comput. Geom. Theory Appl.*, 1(4):189–201, 1992.
- 7 B. Aronov and M. Sharir. On translational motion planning of a convex polyhedron in 3-space. *SIAM J. Comput.*, 26(6):1785–1803, 1997.
- 8 S. Arya and T. M. Chan. Better ε -dependencies for offline approximate nearest neighbor search, Euclidean minimum spanning trees, and ε -kernels. In *Proc. 30th Annu. Sympos. Comput. Geom.*, pages 416–425, 2014.
- 9 S. Arya, G. D. da Fonseca, and D. M. Mount. Near-optimal ε -kernel construction and related problems. In *Proc. 33rd Internat. Sympos. Comput. Geom.*, pages 10:1–15, 2017. URL: <https://arxiv.org/abs/1604.01175>.
- 10 S. Arya, G. D. da Fonseca, and D. M. Mount. On the combinatorial complexity of approximating polytopes. *Discrete Comput. Geom.*, 58(4):849–870, 2017.
- 11 S. Arya, G. D. da Fonseca, and D. M. Mount. Optimal approximate polytope membership. In *Proc. 28th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 270–288, 2017.
- 12 S. Arya, G. D. da Fonseca, and D. M. Mount. Approximate polytope membership queries. *SIAM J. Comput.*, 47(1):1–51, 2018.
- 13 L. Barba and S. Langerman. Optimal detection of intersections between convex polyhedra. In *Proc. 26th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 1641–1654, 2015.
- 14 G. Barequet and S. Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. *J. Algorithms*, 38(1):91–109, 2001.
- 15 J.-D. Boissonnat, E. De Lange, and M. Teillaud. Minkowski operations for satellite antenna layout. In *Proc. 13th Annu. Sympos. Comput. Geom.*, pages 67–76, 1997.
- 16 E. M. Bronshteyn and L. D. Ivanov. The approximation of convex sets by polyhedra. *Siberian Math. J.*, 16:852–853, 1976.
- 17 T. M. Chan. Approximating the diameter, width, smallest enclosing cylinder, and minimum-width annulus. *Internat. J. Comput. Geom. Appl.*, 12:67–85, 2002.
- 18 T. M. Chan. Faster core-set constructions and data-stream algorithms in fixed dimensions. *Comput. Geom. Theory Appl.*, 35(1):20–35, 2006.
- 19 T. M. Chan. Applications of Chebyshev polynomials to low-dimensional computational geometry. In *Proc. 33rd Internat. Sympos. Comput. Geom.*, pages 26:1–15, 2017.
- 20 B. Chazelle. An optimal algorithm for intersecting three-dimensional convex polyhedra. *SIAM J. Comput.*, 21(4):671–696, 1992.
- 21 B. Chazelle and D. P. Dobkin. Detection is easier than computation. In *Proc. 12th Annu. ACM Sympos. Theory Comput.*, pages 146–153, 1980.
- 22 B. Chazelle and D. P. Dobkin. Intersection of convex objects in two and three dimensions. *J. Assoc. Comput. Mach.*, 34:1–27, 1987.
- 23 B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. *J. Algorithms*, 21:579–597, 1996.
- 24 M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 3rd edition, 2010.
- 25 D. P. Dobkin and D. G. Kirkpatrick. Fast detection of polyhedral intersection. *Theo. Comp. Sci.*, 27(3):241–253, 1983.

- 26 D. P. Dobkin and D. G. Kirkpatrick. Determining the separation of preprocessed polyhedra—A unified approach. In *Proc. Internat. Colloq. Automata Lang. Prog.*, pages 400–413, 1990.
- 27 R. M. Dudley. Metric entropy of some classes of sets with differentiable boundaries. *J. Approx. Theory*, 10(3):227–236, 1974.
- 28 C. A. Duncan, M. T. Goodrich, and E. A. Ramos. Efficient approximation and optimization algorithms for computational metrology. In *Proc. Eighth Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 121–130, 1997.
- 29 E. Fogel, D. Halperin, and C. Weibel. On the exact maximum complexity of Minkowski sums of polytopes. *Discrete Comput. Geom.*, 42(4):654–669, 2009.
- 30 X. Guo, L. Xie, and Y. Gao. Optimal accurate Minkowski sum approximation of polyhedral models. *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues*, pages 179–188, 2008.
- 31 D. Halperin, O. Salzman, and M. Sharir. Algorithmic motion planning. In J. E. Goodman, J. O’Rourke, and C. D. Tóth, editors, *Handbook of Discrete and Computational Geometry*, Discrete Mathematics and its Applications. CRC Press, 2017.
- 32 S. Har-Peled, T. M. Chan, B. Aronov, D. Halperin, and J. Snoeyink. The complexity of a single face of a Minkowski sum. In *Proc. Seventh Canad. Conf. Comput. Geom.*, pages 91–96, 1995.
- 33 P. Jiménez, F. Thomas, and C. Torras. 3D collision detection: A survey. *Computers & Graphics*, 25(2):269–285, 2001.
- 34 F. John. Extremum problems with inequalities as subsidiary conditions. In *Studies and Essays Presented to R. Courant on his 60th Birthday*, pages 187–204. Interscience Publishers, Inc., New York, 1948.
- 35 A. Kaul and J. Rossignac. Solid-interpolating deformations: construction and animation of pips. *Computers & graphics*, 16(1):107–115, 1992.
- 36 M. Lin and S. Gottschalk. Collision detection between geometric models: A survey. In *Proc. of IMA conference on mathematics of surfaces*, volume 1, pages 602–608, 1998.
- 37 D. M. Mount. Geometric intersection. In J. E. Goodman, J. O’Rourke, and C. D. Tóth, editors, *Handbook of Discrete and Computational Geometry*, Discrete Mathematics and its Applications. CRC Press, 2017.
- 38 D. E. Muller and F. P. Preparata. Finding the intersection of two convex polyhedra. *Theo. Comp. Sci.*, 7(2):217–236, 1978.
- 39 J. O’Rourke. *Computational geometry in C*. Cambridge University Press, 1998.
- 40 L. Pachter and B. Sturmfels. *Algebraic statistics for computational biology*, volume 13. Cambridge University Press, 2005.
- 41 R. Schneider. *Convex bodies: The Brunn-Minkowski theory*. Cambridge University Press, 1993.
- 42 M. I. Shamos. Geometric complexity. In *Proc. Seventh Annu. ACM Sympos. Theory Comput.*, pages 224–233, 1975.
- 43 H. R. Tiwary. On the hardness of computing intersection, union and Minkowski sum of polytopes. *Discrete Comput. Geom.*, 40(3):469–479, 2008.
- 44 G. Varadhan and D. Manocha. Accurate Minkowski sum approximation of polyhedral models. *Graphical Models*, 68(4):343–355, 2006.
- 45 H. Yu, P. K. Agarwal, R. Poredy, and K. R. Varadarajan. Practical methods for shape fitting and kinetic data structures using coresets. *Algorithmica*, 52(3):378–402, 2008.