Penetration Depth of Two Convex Polytopes in 3D*

Pankaj K. Agarwal† Leonidas J. Guibas‡ Sariel Har-Peled§
Alexander Rabinovitch¶ Micha Sharir‖

October 17, 2000

Abstract

Let A and B be two convex polytopes in $\mathbb{R}^3$ with m and n facets, respectively. The *penetration depth* of A and B, denoted as $\pi(A, B)$, is the minimum distance by which A has to be translated so that A and B do not intersect. We present a randomized algorithm that computes $\pi(A, B)$ in $O(m^{3/4+\varepsilon}n^{3/4+\varepsilon} + m^{1+\varepsilon} + n^{1+\varepsilon})$ expected time, for any constant $\varepsilon > 0$. It also computes a vector t such that $\|t\| = \pi(A, B)$ and $\mathrm{int}(A + t) \cap B = \emptyset$. We show that if the Minkowski sum $B \oplus (-A)$ has K facets, then the expected running time of our algorithm is $O\left(K^{1/2+\varepsilon}m^{1/4}n^{1/4} + m^{1+\varepsilon} + n^{1+\varepsilon}\right)$, for any $\varepsilon > 0$.

We also present an approximation algorithm for computing $\pi(A, B)$. For any $\delta > 0$, we can compute, in time $O(m + n + (\log^2(m + n))/\delta)$, a vector t such that $\|t\| \leq (1+\delta)\pi(A, B)$ and $\mathrm{int}(A+t)\cap B = \emptyset$. Our result also gives a δ-approximation algorithm for computing the width of A in time $O(n+(1/\delta)\log^2(1/\delta))$, which is simpler and faster than the recent algorithm by Chan [5].

*Work by P.A. was supported by Army Research Office MURI grant DAAH04-96-1-0013, by a Sloan fellowship, by NSF grants EIA–9870724, and CCR–9732787, and by a grant from the U.S.-Israeli Binational Science Foundation. Work by L.G. was supported in part by National Science Foundation grant CCR–9623851 and by US Army MURI grant 5–23542–A. Work by S.H.-P. was supported by Army Research Office MURI grant DAAH04-96-1-0013. Work by M.S. was supported by NSF Grants CCR-97-32101, CCR-94-24398, by grants from the U.S.-Israeli Binational Science Foundation, the G.I.F., the German-Israeli Foundation for Scientific Research and Development, and the ESPRIT IV LTR project No. 21957 (CGAL), and by the Hermann Minkowski–MINERVA Center for Geometry at Tel Aviv University.

†Center for Geometric Computing, Department of Computer Science, Box 90129, Duke University, Durham, NC 27708-0129, USA. E-mail: pankaj@cs.duke.edu

‡Computer Graphics Laboratory, Computer Science Department, Stanford University, Stanford CA 94305 E-mail: guibas@cs.stanford.edu

§Center for Geometric Computing, Department of Computer Science, Box 90129, Duke University, Durham, NC 27708-0129, USA. Current address: Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801-2987. E-mail: sariel@uiuc.edu

¶Synopsys Inc., 154 Crane Meadow Rd, Suite 300, Marlboro, MA 01752, USA. E-mail: alexra@synopsys.com

‖School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel; and Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA. E-mail: sharir@math.tau.ac.il

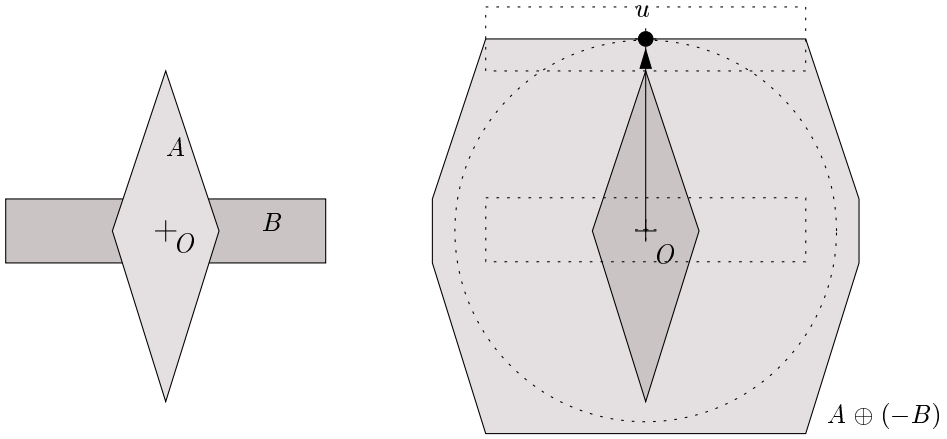

Figure 1: (a) Two convex polygons A and B. (b) Minkowski sum $P = A \oplus (-B)$; u is the closest point from O to ∂P, so $\pi(A, B) = \|Ou\|$.

1 Introduction

Let A and B be two convex polytopes in $\mathbb{R}^3$, with m and n facets, respectively. The *penetration depth* of A and B, denoted as $\pi(A, B)$, is defined as

$$\pi(A, B) = \min\{\|t\| \mid \text{int}(A + t) \cap B = \emptyset,\ t \in \mathbb{R}^3\}.$$

See Figure 1. One of the motivations for this problem comes from the field of robotics. Consider, for instance, the problem of collision detection in robot motion planning, where distance between objects is measured in the Euclidean metric. Numerous efficient algorithms are known for computing the minimum distance between two polyhedra in two and three dimensions (see [8, 11]). Whenever two objects intersect, this distance measure is zero. Thus, it fails to provide any information about the *extent of penetration*. The penetration depth is a useful and natural measure of this extent [4, 14]. In addition, penetration depth can be a useful quantity to have available during physical simulations. Such simulations sample a moving system during discrete time steps and detect collisions between objects using a variety of methods. When a collision is detected, a penetration has usually occurred, because of the discrete time sampling. The penetration depth of the colliding bodies can be very useful in computing how to roll the simulation back to the instant of first contact, and in estimating the impulse force required for the appropriate collision response.

The problem is closely related to that of computing the *width* of a convex polytope A. The width of A is the shortest distance between any pair of parallel planes that support A. We will note below that $\pi(A, A) = \text{width}(A)$. Thus the penetration depth is a natural extension of width. The best algorithm known for computing the width is by Agarwal and Sharir [2]; it is a randomized algorithm that runs in $O(n^{3/2+\varepsilon})$ expected time, for any constant $\varepsilon > 0$. Their algorithm is based on a randomized algorithm, presented in [2], for computing the closest bichromatic pair of lines for two "vertically-separated" sets L and L'

of lines in $\mathbb{R}^3$ (see below for a more precise definition), in expected time $O(|L|^{3/4+\varepsilon}|L'|^{3/4+\varepsilon} + |L|^{1+\varepsilon} + |L'|^{1+\varepsilon})$, for any $\varepsilon > 0$. We use their closest-pair algorithm for computing $\pi(A, B)$ in expected time $O(m^{3/4+\varepsilon}n^{3/4+\varepsilon} + m^{1+\varepsilon} + n^{1+\varepsilon})$, for any $\varepsilon > 0$. Actually, we will show that if the number of facets of the Minkowski sum $B \oplus (-A)$ is K, then the expected running time of the algorithm is $O(K^{1/2+\varepsilon}m^{1/4}n^{1/4} + m^{1+\varepsilon} + n^{1+\varepsilon})$, for any $\varepsilon > 0$. This is, to the best of our knowledge, the first subquadratic algorithm for computing $\pi(A, B)$.

Dobkin et al. [10] showed that A and B can be preprocessed in $O(m + n)$ time so that, for a direction u, the distance by which A has to be translated in direction u to separate it from B, denoted as $\Delta(u)$, can be computed in $O(\log^2(m + n))$ time. We use this result to obtain a simple approximation algorithm for computing $\pi(A, B)$. In particular, for any given $\delta > 0$, we present an $O(m + n + (\log^2(m + n))/\delta)$-time algorithm for computing a vector t such that $\text{int}(A + t) \cap B = \emptyset$ and $\|t\| \leq (1 + \delta)\pi(A, B)$.

Our results imply an "output-sensitive" algorithm or computing the width of a convex polytope A with n facets in randomized expected time $O(K^{1/2+\varepsilon}n^{1/2}+n^{1+\varepsilon})$, where K is the number of facets in $A\oplus(-A)$, and a $(1+\delta)$-approximation algorithm for computing the width of A in time $O(n+(1/\delta)(\log^2(1/\delta))$. This approximation algorithm is simpler and faster than the recent algorithm by Chan [5], which computes a $(1 + \delta)$-approximation of width(A) in time $O(n+(\log^c n)/\delta)$ for some constant $c > 2$. We also present an $O(m+n+k\log^2(m+n))$-time algorithm for computing the exact penetration depth under any polyhedral metric, where k is the number of vertices in the polytope defining the metric. Finally, we present an alternative algorithm that is likely to be more efficient when the penetration is shallow. Both of these algorithms are based on the technique used in the approximation algorithm.

2 Computing the Penetration Depth

2.1 Penetration depth and width

Before describing the algorithm for computing $\pi(A, B)$, we note the relationship between the penetration depth of two polytopes and the width of a polytope.

Proposition 2.1 *For any convex polytope P in $\mathbb{R}^3$,* width$(P) = \pi(P, P)$.

Proof: Let λ denote the length of the shortest translation vector that separates two initially-identical copies of P. Let v be a vector realizing the width of P; that is, v is a shortest vector for which there exists a plane h such that P lies between h and $h+v$. Clearly, $\text{int}(P+v)\cap\text{int}(P) = \emptyset$, and therefore $\lambda \leq \|v\| = \text{width}(P)$. As for the other direction, let u be a shortest separating translation vector. Clearly, P and $P+u$ touch each other but have disjoint interiors. Thus, there is a plane H that separates the interiors of P and $P + u$, and intersects both P and $P + u$. In particular, P lies between the two planes H and $H - u$. Since the distance between H and $H - u$ is $\leq \|u\|$, it follows that

$$\text{width}(P) \leq d(H, H - u) \leq \|u\|.$$

■

This proposition suggests that we attempt to modify the width algorithm by Agarwal and Sharir [2] to compute $\pi(A, B)$, which is indeed what we proceed to do. Conversely, we will also specialize the new techniques developed in this paper to obtain new approximation and output-sensitive algorithms for computing width(A).

2.2 A general exact algorithm

Let A and B be two convex polytopes as defined above. Using linear programming, we can determine in $O(m+n)$ time whether A and B intersect [9]. If A and B do not intersect, then we set $\pi(A, B) = 0$ and stop. So we assume that $A \cap B \neq \emptyset$. We also assume that the vertices of A and B are in general position. There are standard techniques, e.g., those based on perturbations, to handle situations where this assumption does not hold.

We can formulate the problem of computing $\pi(A, B)$ in terms of the *configuration space* that represents all possible placements of A relative to (the fixed) B. That is, A turns into a point $p(A)$ and B turns into the *Minkowski sum* $B \oplus (-A) = \{x - y \mid x \in B, y \in A\}$. Let us assume that the initial location of the point $p(A)$ corresponding to A in the configuration space is the origin O of the coordinate system. Note that $p(A)$ is inside the polytope $\mathcal{P} = B \oplus (-A)$ if and only if A (in the corresponding translated placement) and B intersect. By construction, it follows that

$$\pi(A, B) = \min\{d(O, x) \mid x \in \partial\mathcal{P}\}.$$

Let x be a point on the boundary of $\mathcal{P}$ so that $d(x, O) = d(O, \mathcal{P})$. Then $\overrightarrow{Ox}$ is orthogonal to the facet of $\mathcal{P}$ containing x. Otherwise, we could obtain an even shorter distance from O to $\partial\mathcal{P}$, which is impossible. Therefore, $d(x, O)$ is attained as a shortest distance between O and a plane that contains the corresponding facet of $\mathcal{P}$. In particular, we can compute the penetration distance by computing the distance between the origin and all the planes that support a facet of $\mathcal{P}$.

Every facet of $\mathcal{P}$ is attained as a Minkowski sum of the form $g \oplus (-f)$, where g is a facet, edge, or vertex of B and f is, respectively, a vertex, edge, or facet of A. It is well known that there are only $O(m+n)$ facets of $\mathcal{P}$ for which g is a facet or a vertex of B (and f is a vertex or a facet of A), and they can all be found in $O((m+n)\log(m+n))$ time (see e.g. [7]). Hence, determining the minimum distance from O to all these facets can be done in near-linear time. The problem is to handle facets that are of the form $e \oplus (-e')$ such that e is an edge of B and e' is an edge of A. In the worst case, there can be $\Omega(mn)$ such facets. However, not every such pair necessarily generates a facet of $\mathcal{P}$.

We construct a family of pairs of subsets of edges $\mathcal{F} = \{(A_1, B_1), \ldots, (A_u, B_u)\}$ such that the following five conditions hold.

(C1) A_i (resp. B_i) is a subset of the edges of A (resp. B).

(C2) Every pair $(e', e) \in A_i \times B_i$ generates a facet of $\mathcal{P}$.

(C3) Every pair of edges that generate a facet of $\mathcal{P}$ appears in some $A_j \times B_j$.

(C4) For each i, the lines supporting the edges in A_i and those in B_i are vertically separated. That is, either all lines supporting the edges of A_i lie above all lines supporting the edges of B_i, or all of them lie below the lines supporting the edges of B_i.

(C5) $\mathcal{F}$ can be partitioned into two subfamilies $\mathcal{F}^A$ and $\mathcal{F}^B$ such that

(i) for every $0 \leq i \leq \lfloor \log_2 m \rfloor$, there are $O((m/2^i) \log m)$ pairs (A_j, B_j) in $\mathcal{F}^A$ for which $2^i \leq |A_j| < 2^{i+1}$. Let $\mathcal{F}_i^A$ denote the subset of these pairs. Then $\sum_{(A_j,B_j)\in\mathcal{F}_i^A} |B_j| = O(n \log n)$; and

(ii) for every $0 \leq i \leq \lfloor \log_2 n \rfloor$, there are $O((n/2^i) \log n)$ pairs (A_j, B_j) in $\mathcal{F}^B$ for which $2^i \leq |B_j| < 2^{i+1}$. Let $\mathcal{F}_i^B$ denote the subset of these pairs. Then $\sum_{(A_j,B_j)\in\mathcal{F}_i^B} |A_j| = O(m \log m)$.

Note that condition (C5) implies that

$$\sum_{i=1}^{u} (|A_i| + |B_i|) = O((m+n) \log^2(m+n)). \tag{1}$$

Suppose we have such a decomposition at our disposal. Then we can compute $\pi(A, B)$ as follows. Recall that our goal is to compute the minimum distance from the origin to the planes supporting the faces of $\mathcal{P}$.

Algorithm: PENETRATION-DEPTH (A, B)

1. For each pair (f, g) such that f is a vertex or facet of A and g is a facet or vertex of B, and $g \oplus (-f)$ is a facet of $\mathcal{P}$, compute the distance from the origin to the plane containing $g \oplus (-f)$. Let Δ^* be the minimum of these distances.

2. For each pair (A_i, B_i) in the above decomposition, find the minimum distance Δ_i from the origin to an element in the set of planes

$$H_i = \{\text{aff}(e \oplus (-e')) \mid e \in B_i, e' \in A_i\},$$

where $\text{aff}(e \oplus (-e'))$ is the plane containing the facet of $B \oplus (-A)$ induced by e and e'.

3. Return $\min\{\Delta^*, \min_i\{\Delta_i\}\}$.

The correctness of this algorithm is obvious. Step 1 considers all facets of $\mathcal{P}$ induced by a vertex-facet pair of A and B. By Condition (C2), the algorithm considers only those pairs of edges that generate facets of $\mathcal{P}$, and by Condition (C3), the algorithm considers all such pairs. It thus suffices to show how to compute Δ_i, for each pair (A_i, B_i), and how to construct the family $\mathcal{F}$.

Computing Δ_i. Let (A_i, B_i) be a pair in $\mathcal{F}$. Denote by L_i and L_i', respectively, the sets of lines that contain the edges of B_i and A_i.

Lemma 2.2 *For any pair $(A_i, B_i) \in \mathcal{F}$, $\Delta_i = d(L_i, L_i')$.*

Proof: Let $e \in B_i$ and $e' \in A_i$, and let ℓ and ℓ' be the lines that contain e and e', respectively. Consider the plane $h = \ell \oplus (-\ell') = \{x - y \mid x \in \ell, y \in \ell'\}$. Note that for any two sets X and Y,

$$d(O, X \oplus (-Y)) = \inf\{||x - y|| \mid x \in X, y \in Y\} = d(X, Y).$$

Therefore $d(O, h) = d(O, \ell \oplus (-\ell')) = d(\ell, \ell')$. Thus,

$$\begin{aligned} \Delta_i &= \min_{h \in H_i} d(O, h) \\ &= \min\{d(O, \ell \oplus (-\ell')) \mid \ell \in L_i, \ell' \in L_i'\} \\ &= \min\{d(\ell, \ell') \mid \ell \in L_i, \ell' \in L_i'\} \\ &= d(L_i, L_i'). \end{aligned}$$

■

By the above lemma, computing Δ_i reduces to computing a closest bichromatic pair of lines in $L_i \times L_i'$. Recall that by Condition (C4) on $\mathcal{F}$, the lines in L_i and L_i' are vertically separated. Agarwal and Sharir [2] showed that under this condition, the closest pair in $L_i \times L_i'$ can be computed in expected time $O(|L_i|^{3/4+\varepsilon}|L_i'|^{3/4+\varepsilon} + |L_i|^{1+\varepsilon} + |L_i'|^{1+\varepsilon})$. Summing this bound over all pairs in $\mathcal{F}$ and using property (C5), routine calculation yields that the total time spent in computing all the Δ_i's is $O(m^{3/4+\varepsilon}n^{3/4+\varepsilon} + m^{1+\varepsilon} + n^{1+\varepsilon})$, for any $\varepsilon > 0$.

Computing $\mathcal{F}$. Our decomposition is based on the following observation. Let $\mathcal{M}$ denote the *Gaussian diagram* (or *normal diagram*) of B. $\mathcal{M}$ is a spherical map on the unit sphere $\mathbb{S}^2$. The vertices of $\mathcal{M}$ are points on $\mathbb{S}^2$, each representing the direction of the outward normal of a facet of B, the edges of $\mathcal{M}$ are great circular arcs, each being the locus of the outward normal directions of all planes supporting B at some fixed edge, and the faces of $\mathcal{M}$ are regions, each being the locus of outward normal directions of all planes supporting B at a vertex. $\mathcal{M}$ can be computed in linear time from B. Let $\mathcal{M}'$ be the similarly-defined normal diagram of $-A$. Consider the superposition of $\mathcal{M}$ and $\mathcal{M}'$. Each intersection point between an arc of $\mathcal{M}$ and an arc of $\mathcal{M}'$, representing respectively an edge e of B and an edge e' of A, gives us a direction $\mathbf{u}$ which is orthogonal to the plane containing the Minkowski sum $e \oplus (-e')$. Furthermore, $e \oplus (-e')$ is a real facet of $B \oplus (-A)$. It follows that a pair of edges of A and B generates a face of $B \oplus (-A)$ if and only if the corresponding arcs intersect in the overlapped diagram. Note that the number of such arc intersections on this diagram can be $\Omega(mn)$.

Our goal is thus to decompose the set of all pairs of intersecting arcs of $\mathcal{M}$ and $\mathcal{M}'$. Without loss of generality, assume that no intersection point of $\mathcal{M}$ and $\mathcal{M}'$ lies on the equator. (We can either handle these intersections separately, or perform a random simultaneous

rotation on $\mathcal{M}$ and $\mathcal{M}'$.) If an arc of $\mathcal{M}$ or $\mathcal{M}'$ crosses the equator, we split it into two by adding a vertex on the arc at the equator. Hence each arc lies completely in the upper or the lower hemisphere. Let $\mathbb{H}$ denote the upper hemisphere of $\mathbb{S}^2$. We will describe how we decompose the set of edges of A and B whose corresponding arcs intersect in $\mathbb{H}$; the lower hemisphere is handled similarly.

Note that the arcs in $\mathcal{M}$ (and in $\mathcal{M}'$) are pairwise disjoint. We centrally project the arcs of $\mathcal{M}$ and $\mathcal{M}'$ that lie in $\mathbb{H}$ onto the plane $h : z = 1$. Since each arc of $\mathcal{M}$ and $\mathcal{M}'$ is a portion of a great circle, it projects to a segment (or a ray) on h. Let E (resp. E') be the set of projected segments of arcs in $\mathcal{M}$ (resp. $\mathcal{M}'$). By construction, the interiors of the segments in E (or E') are pairwise disjoint.

As described in [6], we decompose the set of intersecting pairs of segments in E and E' into a family $\mathcal{F}' = \{(E_1, E'_1), \ldots, (E_u, E'_u)\}$ as follows. We construct two segment trees T_A and T_B on the segments of E and E', respectively. Each node v of T_A (resp. T_B) corresponds to a vertical strip, with an associated subset $E_v \subseteq E$ (resp. $E'_v \subseteq E'$) that completely cross the strip. For each such subset, we construct a balanced binary tree, sorted by the height of those segments inside the strip (the segments do not intersect, and thus the ordering is well defined). For each node w of this binary tree, we refer to the subset of segments stored in the subtree rooted at w as a *canonical* subset.

For each segment e of E' (resp. E), we find the nodes v of T_A (resp. T_B) such that at least one endpoint of e lies inside the strip associated with the parent of v; there is a logarithmic number of such nodes. We report all segments of E_v (resp. E'_v) intersected by the segment as the union of a logarithmic number of canonical subsets. After repeating this step for all segments, for each canonical subset E_w of T_A, we report the pair (E_w, E'_w), where E'_w is the subset of segments for which the query procedure returned E_w as one of the canonical subsets. We do the same for the canonical subsets of T_B. It is shown in [7] that if segments $e \in E, e' \in E'$ intersect, then there is at least one such pair (E_z, E'_z) such that $e \in E_z$ and $e' \in E'_z$, and that the total time spent and storage used is $O((m+n)\log(m+n))$. (Note that an intersecting pair (e, e') may be reported twice in this algorithm—once when searching with e and once when searching with e'.) Finally, for each pair (E_w, E'_w), let A_w (resp. B_w) be the set of corresponding edges of A and B. We add the pair (A_w, B_w) to $\mathcal{F}$. $\mathcal{F}^A$ (resp. $\mathcal{F}^B$) is the subset of pairs corresponding to the canonical subsets of T_A (resp. T_B). The argument in [7] shows that $\mathcal{F}$ satisfies conditions (C1)–(C3) and (C5). Condition (C4) follows from the following lemma.

Lemma 2.3 *Let e be an edge of B and e' an edge of A such that the corresponding arcs intersect in $\mathbb{H}$. Then the line supporting e lies above the line supporting e'.*

Proof: Since the arcs corresponding to e and e' intersect in $\mathbb{H}$, the sum $e \oplus (-e')$ is a facet of $B \oplus (-A)$ with an outward normal direction u that points upwards. By construction of the diagrams, there are planes h, h' orthogonal to u and supporting, respectively, B at e and A at e'. Moreover, relative to the direction u, B lies below h and A lies above h'. It follows that since A and B intersect, the plane h is above the plane h' relative to the direction u. Thus also the line ℓ containing e is above the line ℓ' containing e' relative to the

direction u. Recall that we assumed that the vertices of A and B are in general position, so, in particular, there are no four coplanar vertices. Thus the lines ℓ and ℓ' are not parallel. Let ℓ_0 be the unique upward-directed vertical line that passes through ℓ and ℓ'. Since the angle between ℓ_0 and u is smaller that $\pi/2$, and a line in direction u crosses h' before h, it follows that ℓ_0 also crosses h' (at a point on ℓ') before it crosses h (at a point on ℓ). Hence ℓ lies vertically above ℓ', as claimed. ■

Hence, we conclude the following.

Theorem 2.4 *Given two convex polytopes A, B in $\mathbb{R}^3$ with m and n vertices, respectively, the penetration depth of A and B can be computed in randomized expected time $O(m^{3/4+\varepsilon}n^{3/4+\varepsilon} + m^{1+\varepsilon} + n^{1+\varepsilon})$, for any $\varepsilon > 0$; the constant of proportionality depends on ε.*

2.3 An output-sensitive bound

Let K denote the number of facets in $\mathcal{P} = B \oplus (-A)$. We derive a bound on the expected running time of the algorithm that depends on K. Note that the pair (A_i, B_i) contributes $|A_i| \cdot |B_i|$ facets to $\mathcal{P}$ by Condition (C2). The expected running time of the algorithm is

$$\sum_{i=1}^{u} O\left((|A_i||B_i|)^{3/4+\varepsilon} + |A_i|^{1+\varepsilon} + |B_i|^{1+\varepsilon}\right) \quad = \quad O\left(K^{\varepsilon}\sum_{i=1}^{u}(|A_i||B_i|)^{3/4} + m^{1+\varepsilon} + n^{1+\varepsilon}\right),$$

where we have used (1) in bounding the sums of the second and third terms.

We obtain a bound on $\sum_{(A_j,B_j)\in\mathcal{F}^A}(|A_j||B_j|)^{3/4}$. A similar argument bounds the quantity for pairs in $\mathcal{F}^B$. Let $K_i = \sum_{(A_j,B_j)\in\mathcal{F}_i^A}|A_j||B_j|$ be the number of facets contributed by the pairs in $\mathcal{F}_i^A$. Recall that $\left|\mathcal{F}_i^A\right| = O((m/2^i)\log m)$. Using Hölder's inequality, we obtain:

$$\begin{aligned}
\sum_{(A_j,B_j)\in\mathcal{F}^A}(|A_j||B_j|)^{3/4} &= \sum_{i=0}^{\log_2 m}\sum_{(A_j,B_j)\in\mathcal{F}_i^A}(|A_j||B_j|)^{3/4} \\
&\le \sum_{i=0}^{\log_2 m}\left(\sum_{(A_j,B_j)\in\mathcal{F}_i^A}|A_j||B_j|\right)^{3/4}\left|\mathcal{F}_i^A\right|^{1/4} \\
&\le \sum_{i=0}^{\log_2 m} O\left(K_i^{3/4}\left(\frac{m\log m}{2^i}\right)^{1/4}\right) \\
&\le O\left((m\log m)^{1/4}\sum_{i=0}^{\log_2 m}\frac{K_i^{3/4}}{2^{i/4}}\right).
\end{aligned}$$

On the other hand, $K_i \le 2^{i+1}\sum_{(A_j,B_j)\in\mathcal{F}_i^A}|B_j| \le c2^i n\log n$ for a constant $c > 1$. The term $\sum_i K_i^{3/4}/2^{i/4}$ is therefore maximized when $K_i = c2^i n\log n$ for $0 \le i \le \log_2\frac{K}{cn\log n}$ and 0

otherwise. Hence,

$$\sum_{i=0}^{\log_2 m} \frac{K_i^{3/4}}{2^{i/4}} = \sum_{i=0}^{\log_2 \frac{K}{cn\log n}} O(2^{i/2}(n\log n)^{3/4}) = O(\sqrt{K}(n\log n)^{1/4}).$$

Therefore $\sum_{(A_j,B_j)\in\mathcal{F}^A} (|A_j||B_j|)^{3/4} = O(\sqrt{K}(mn\log m\log n)^{1/4})$, and the same bound holds for the sum over pairs in $\mathcal{F}^B$. We thus obtain the following.

Theorem 2.5 *Given two intersecting convex polytopes A, B in $\mathbb{R}^3$ with m and n vertices, respectively, such that $B \oplus (-A)$ has K facets, one can compute the penetration depth of A and B in randomized expected time $O\left(K^{1/2+\varepsilon}m^{1/4}n^{1/4} + m^{1+\varepsilon} + n^{1+\varepsilon}\right)$ for any $\varepsilon > 0$.*

An immediate corollary of the above theorem is the following.

Corollary 2.6 *Given a convex polytope A in $\mathbb{R}^3$ with n vertices such that $A \oplus (-A)$ has K facets, one can compute the width of A in randomized expected time $O\left(K^{1/2+\varepsilon}n^{1/2} + n^{1+\varepsilon}\right)$ for any $\varepsilon > 0$.*

3 An Approximation Algorithm

We now present an efficient algorithm for approximating the penetration depth of A and B. That is, for a given $\delta > 0$, the algorithm computes a translation vector t such that the interiors of $A + t$ and B are disjoint and $\|t\| \leq (1+\delta)\pi(A, B)$. The algorithm is as follows.

Algorithm: Approx-Separation $(A,\ B)$

1. Define on the unit sphere of directions a grid G of points in the following manner: Divide the interval of angles $[0, \pi]$ into $\left\lceil c_1/\sqrt{\delta}\right\rceil$ subintervals of equal length, delimited by the points $0 = p_0, p_1, \cdots, p_{\lceil c_1/\sqrt{\delta}\rceil} = \pi$, where c_1 is a constant independent of δ. Then the grid G is defined as the set of points

$$G = \left\{ (p_i, 2p_j) \mid 0 \leq i, j \leq \left\lceil c_1/\sqrt{\delta}\right\rceil \right\},$$

   where the points are given in spherical coordinates (φ, θ). The constant c_1 is chosen so that the spherical distance from any point on the sphere to its nearest grid point is at most $\sqrt{\delta}$.

2. For each point $p \in G$, perform the following ray-shooting query: Find the intersection point on the boundary of $B \oplus (-A)$ with the ray $\overrightarrow{Op}$. Let $\Delta(p)$ be the Euclidean distance from O to the boundary of $B \oplus (-A)$ in this direction. We will explain below how the ray-shooting can be performed efficiently without explicit computation of the Minkowski sum.

3. Output $\Delta = \min_{p\in G}\{\Delta(p)\}$ as an approximate solution.

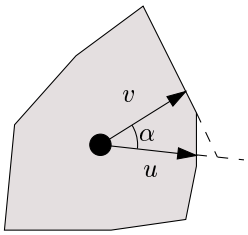

Figure 2: Illustration of the proof of Lemma 3.1

Lemma 3.1 *For any $\delta > 0$, algorithm* Approx-Separation *computes correctly a translation of length Δ that separates A and B, such that $\Delta \leq (1+\delta)\pi(A,B)$.*

Proof: Let v be the vector that realizes the penetration depth $\pi(A,B)$. Let u be the vector computed by the algorithm, as it ray-shoots in the direction which is closest (in angular distance) to that of v. Clearly, the vector u, lying on the boundary of $B \oplus (-A)$, separates A and B. Let α be the angle between v and u. By construction, $\alpha \leq \sqrt{\delta}$. See Figure 2.

It is easy to verify that

$$||u|| \leq \frac{||v||}{\cos\alpha} \leq \frac{||v||}{1-\alpha^2/2} \leq (1+\alpha^2)||v|| \leq (1+\delta)||v||.$$

■

The size of the grid built by the Approx-Separation algorithm is $O(1/\delta)$. It was shown by Dobkin et al. [10], that after a linear-time preprocessing of A and B into suitable data structures, the shortest separation of A and B along any query direction u can be computed in time $O(\log^2(m+n))$. This operation is equivalent to performing a ray shooting in the direction u from the origin toward $\partial\mathcal{P}$. Therefore, the total running time of the algorithm Approx-Separation is $O(m+n+(\log^2(m+n))/\delta)$. We have thus shown:

Theorem 3.2 *Given two convex polytopes A and B in $\mathbb{R}^3$, with m and n facets, respectively, and a parameter $\delta > 0$, one can compute, in time $O(m+n+(\log^2(m+n))/\delta)$, a separating translation for A and B whose length is at most $(1+\delta)\pi(A,B)$.*

Applying Proposition 2.1, we also obtain the following corollary:

Corollary 3.3 *For any $\delta > 0$, a $(1+\delta)$-approximation of the width of a convex polytope in $\mathbb{R}^3$ with n facets can be computed in time $O(n+(\log^2 n)/\delta)$.*

The above result can be further improved as follows: For a convex polytope P with n vertices in $\mathbb{R}^3$, we first compute in linear time a transformation T that map P into a "fat" convex polytope [3], and then compute in $O(n+(1/\delta)\log(1/\delta))$ time another convex polytope Q', a (δ/c)-approximation of $T(P)$ for an appropriate constant c [1, Theorem 3.3]. Q' has $O(1/\delta)$ vertices, $P \subseteq Q = T^{-1}(Q')$, and width$(Q) \leq (1+\delta/3)$width$(P)$. Finally,

using Corollary 3.3, we compute in $O\left(n + (1/\delta)\log^2(1/\delta)\right)$ time a $(1+\delta/3)$-approximation of the width of Q, which gives a δ-approximation to the width of P. We therefore obtain the following.

Theorem 3.4 *For any $\delta > 0$, a $(1+\delta)$-approximation of the width of a convex polytope in $\mathbb{R}^3$ with n facets can be computed in time $O\left(n + (1/\delta)\log^2(1/\delta)\right)$.*

Remark: As a matter of fact, the Dobkin-Kirkpatrick hierarchical representations of two convex polytopes A and B can be used to obtain efficient implementation of various extremal queries concerning the Minkowski sum $B \oplus (-A)$ without its explicit construction. See [15] for details.

3.1 Penetration depth under polyhedral metrics

Another application of our approximation algorithm is to obtain a linear-time algorithm for computing the penetration depth of of A and B under any polyhedral norm. Let Q be a centrally-symmetric convex polytope with k vertices, and let $||\cdot||_Q$ denote the norm induced by Q. We observe that the $||\cdot||_Q$-distance from O to the boundary of $\mathcal{P}$ is equal to the largest scaling factor λ such that $\lambda Q \subseteq \mathcal{P}$. As is easily seen, a vertex of λQ must then touch $\partial\mathcal{P}$. Moreover, as λ varies, each vertex of λQ traces a ray from the origin. Hence, to find the largest λ, we perform ray-shooting queries from O in each of the k directions of the rays traced by the vertices of Q. For each of these ray-shooting queries, we compute the scaling factor λ that corresponds to the hitting point of that ray with $\partial\mathcal{P}$. The smallest of these values is the desired $||\cdot||_Q$-length of the shortest separating translation. We have thus shown the following.

Corollary 3.5 *Let A and B be two convex polytopes A and B in $\mathbb{R}^3$, with m and n facets, respectively, and let Q be a convex polytope with k vertices. The shortest separating translation of A and B under the polyhedral distance induced by Q can be computed in $O(m+n+k\log^2(m+n))$ time.*

3.2 Handling shallow penetrations

If the penetration of A into B is relatively small, then one might expect that the following combinatorial property holds in practice. Let $\delta > 0$ be a small parameter. Then the number K_δ of facets of $\mathcal{P} = B \oplus (-A)$ whose distance from the origin is at most $(1+\delta)\pi(A,B)$ is small. If this is the case, then the following more efficient algorithm computes $\pi(A,B)$.

Algorithm: SHALLOW-PENETRATION $(A,\ B)$

1. Construct the grid G as in Algorithm APPROX-SEPARATION.
2. Using Algorithm APPROX-SEPARATION, compute a real value Δ such that $\Delta \leq (1+\delta/4)\pi(A,B)$.
3. Compute $G' = \{u \in G \mid \Delta(u) \leq (1+\delta/4)\Delta\}$.
4. Let $\mathcal{B}$ be the ball of radius $(1+\delta/2)\Delta \leq (1+\delta)\pi(A,B)$ centered at the origin.
5. For each $u \in G'$, do the following:
   (i) Compute the face f of $\mathcal{P}$ supported by the plane orthogonal to u.
   (ii) By performing an implicit breadth-first search on $\partial\mathcal{P}$, compute the connected component C_u of $(\partial\mathcal{P}) \cap \mathcal{B}$ that contains f. (If $C_u = C_v$ for two directions $u \neq v$, we compute the connected component C_u only once.)
   (iii) Compute $\Delta_u = \min_{f \in C_u} d(O,f)$, where f is a facet of $\mathcal{P}$ in C_u.
6. Return $\min_{u \in G'} \Delta_u$.

Steps (1)–(3) can be performed in $O(m+n+(\log^2(m+n))/\delta)$ time as described in the algorithm APPROX-SEPARATION. For a given $u \in G'$, we can compute C_u in $O((1+|C_u|)\log(m+n))$ time by locating u in the normal diagrams $\mathcal{M}$ and $\mathcal{M}'$ and by traversing the two diagrams simultaneously. We omit the easy details. Computing Δ_u takes $O(|C_u|)$ time. Since we traverse each connected component of $(\partial\mathcal{P}) \cap \mathcal{B}$ at most once, the total time spent in Step (5) is $O((K_\delta + 1/\delta)\log(m+n))$, where K_δ is the number of facets of $\mathcal{P}$ that lies within distance $(1+\delta)\pi(A,B)$ from O. The same argument as in Lemma 3.1 can be used to show that the above algorithm computes all those connected components of $(\partial\mathcal{P}) \cap \mathcal{B}$ that contain a facet within distance $\pi(A,B)$ from O. Hence, $\min_{u \in G'} \Delta_u = \pi(A,B)$. We thus obtain the following.

Theorem 3.6 *Given two convex polytopes A and B in $\mathbb{R}^3$, with m and n facets, respectively, and a parameter $\delta > 0$, one can compute $\pi(A,B)$ in time $O(m+n+K_\delta \log(m+n)+(\log^2(m+n))/\delta)$, where K_δ is the number of facets of $B \oplus (-A)$ within distance $(1+\delta)\pi(A,B)$ from the origin.*

4 Conclusions

We presented the first subquadratic algorithm for computing the penetration depth of two convex polytopes in $\mathbb{R}^3$, by showing that it is closely related to computing the width of a convex polytope. We also presented simple, near-linear approximation algorithms. We conclude by mentioning two open problems.

(i) Can the penetration depth of two convex polytopes be computed in near-linear time?

(ii) Is there an $o(m^3n^3)$-time algorithm for computing the penetration depth of two *non-convex* polytopes in $\mathbb{R}^3$ with m and n facets, respectively? (An $O(n^6)$-time solution is obtained by explicit construction of the Minkowski sum $B \oplus (-A)$, where A, B are the given polytopes.) Can one develop a fast approximation algorithm in this case? We remark that if the two polytopes are star shaped then their penetration depth can be computed in $O(m^2n^2\alpha(mn))$ time, by constructing the (boundary of the) above sum as the upper envelope of $O(mn)$ triangles, as viewed from a common center point, with respect to which both A and B are star-shaped (using an algorithm similar to that in [13]).

Acknowledgments

The authors thank Boris Aronov for useful discussions and Ming Lin for pointing out a couple of relevant references.

References

[1] P. K. Agarwal, S. Har-Peled, M. Sharir, and K. R. Varadarajan. Approximate shortest paths on a convex polytope in three dimensions. *J. Assoc. Comput. Mach.*, 44:567–584, 1997.

[2] P. K. Agarwal and M. Sharir. Efficient randomized algorithms for some geometric optimization problems. *Discrete Comput. Geom.*, 16:317–337, 1996.

[3] G. Barequet and S. Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. In *Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, pages 82–91, 1999.

[4] S. Cameron. Enhancing gjk: Computing minimum and penetration distance between convex polyhedra. In *Proc. Int. Conf. Robot. Auto.*, pages 3112–3117, 1997.

[5] T.M. Chan. Approximating the diameter, width, smallest enclosing cylinder and minimum-width annulus. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 300–309, 2000.

[6] B. Chazelle, H. Edelsbrunner, L. J. Guibas, and M. Sharir. Diameter, width, closest line pair and parametric searching. *Discrete Comput. Geom.*, 10:183–196, 1993.

[7] B. Chazelle, H. Edelsbrunner, L.J. Guibas, and M. Sharir. Algorithms for bichromatic line segment problems and polyhedral terrains. *Algorithmica*, 11:116–132, 1994.

[8] F. Chin and C. A. Wang. Optimal algorithms for the intersection and the minimum distance problems between planar polygons. *IEEE Trans. Comput.*, C-32(12):1203–1207, 1983.

[9] M. de Berg, M. van Kreveld, M. H. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications.* Springer-Verlag, 1997.

[10] D. Dobkin, J. Hershberger, D. Kirkpatrick, and S. Suri. Computing the intersection-depth of polyhedra. *Algorithmica*, 9:518–533, 1993.

[11] D. P. Dobkin and D. G. Kirkpatrick. A linear algorithm for determining the separation of convex polyhedra. *J. Algorithms*, 6:381–392, 1985.

[12] R. M. Dudley. Metric entropy of some classes of sets with differentiable boundaries. *J. Approx. Theory*, 10(3):227–236, 1974.

[13] H. Edelsbrunner, L.J. Guibas, and M. Sharir. The upper envelope of piecewise linear functions: algorithms and applications. *Discrete Comput. Geom.*, 4:311–336, 1989.

[14] E.G. Gilbert and C.J. Ong. New distances for the separation and penetration of objects. In *Proc. Int. Conf. Robot. Auto.*, pages 579–586, 1994.

[15] A. Rabinovich. Computing the shortest translation separating two convex polytopes in $\Re^3$. M.S. thesis, Dept. Computer Science, Tel Aviv University, Tel Aviv, Israel, 1999.