Efficient Algorithms for Bichromatic Separability

Pankaj K. Agarwal*  Boris Aronov†  Vladlen Koltun‡

Abstract
A closed solid body separates one point set from another if it contains the former and the closure of its complement contains the latter. We present a near-linear algorithm for deciding whether two sets of \( n \) points in 3-space can be separated by a prism, near-quadratic algorithms for separating by a slab or a wedge, and a near-cubic algorithm for separating by a double-wedge. The latter three algorithms improve the previous best known results by an order of magnitude, while the prism separability algorithm constitutes an improvement of two orders of magnitude.

1 Introduction
Many problems in supervised machine learning can be formulated as classifying objects into a finite set of categories, based on a given training set. In its simplest form, each object in the training set is labeled +1 or -1, and the goal is to build a predictor based on this training set that enables us to classify a new object into one of these two classes. Because of its wide range of applications, this fundamental classification problem and its generalizations have been extensively studied in machine learning. Numerous powerful learning techniques such as decision trees, boosting, support vector machines, logistic regression, etc. have been developed in the last two decades. See the recent book by Hastie et al. [13] for a discussion on these and other techniques. Despite these advances in statistical techniques, relatively little progress has been made on combinatorial techniques for classification, especially in three and higher dimensions.

In this paper we study the following simple geometric version of the above classification problem in \( \mathbb{R}^3 \), also known as the separability problem: Let \( R \) and \( B \) be two sets of \( n \) points each in \( \mathbb{R}^3 \). Given a family of closed three-dimensional bodies \( F \), determine whether there exists \( F \in F \) such that \( R \subseteq F \) and \( B \cap \text{int} F = \emptyset \) and if so return such an \( F \). This separability problem is obviously an instance of supervised learning, as the separator \( F \) can be used to predict the class of an arbitrary point in \( \mathbb{R}^3 \). Namely, if a point lies in \( F \), we assign it to \( R \), otherwise to \( B \).

Related work. If \( F \) is the class of all half-spaces, the separability problem can be reduced to linear programming and solved in linear time for any fixed dimension [18], and algorithms are known for finding an "optimal" separating hyperplane [21]. Most of the work in computational geometry on separability/classification has focused on two-dimensional problems. A series of papers studied the problem of separating two sets in the plane by a circle, eventually leading to a linear-time algorithm by O’Rourke et al. [20]. Recently, researchers have studied the problem of separating two planar point sets by other objects such as a convex polygon with the minimum number of edges [10], double wedges [14], and wedges and strips [15]. Arkin et al. [3] proved an \( \Omega(n \log n) \) lower bound for many of these separability problems. The problem of separating two planar sets by a simple polygon with the minimum number of edges is known to be NP-Complete [12] and an approximation algorithm is given by Mitchell [19]. See [2] for a few other related separation results in the plane.

Relatively little is known about the separability problems in three and higher dimensions. The algorithm by O’Rourke extends to higher dimensions. Recently, Hurtado et al. [16] proposed cubic or slightly super-cubic separability algorithms in \( \mathbb{R}^3 \) when \( F \) is the family of slabs (regions bounded by pairs of parallel planes), convex dihedral wedges (intersections of two half-spaces), prisms, and cones. They also developed an \( O(n^3) \) algorithm for deciding separability with a double wedge and algorithms with running times ranging from \( O(n^2) \) to \( O(n^3) \) for other three-dimensional separability problems. Researchers have also studied the problem of separating geometric objects other than points.

---

*Department of Computer Science, Duke University, Durham, NC 27708, USA; pankaju@cs.duke.edu. Supported by NSF under grants CCR-00-80013, EIA-98-70724, EIA-99-72879, EIA-01-31905 and CCR-02-04118, and by a grant from the U.S.-Israel Binational Science Foundation.

†Department of Computer and Information Science, Polytechnic University, Brooklyn, NY 11201-3840, USA; aronov@poly.edu. Supported in part by NSF Grants CCR-00-72968 and ITR CCR-00-81964, and by a grant from the U.S.-Israel Binational Science Foundation. Part of this work on this paper by B.A. was carried out while visiting Mathematical Sciences Research Institute in Berkeley, California.

‡Computer Science Division, University of California, Berkeley, CA 94720-1776, USA; vladlen@cs.berkeley.edu; http://www.cs.berkeley.edu/~vladlen/. Supported by NSF Grant CCR-01-21555.
See [6,7] and references therein for several such results.

**Our results.** In this paper we develop separability algorithms for various families of separators in $\mathbb{R}^3$. In particular, we obtain the following results:

1. An $O(n \log^4 n)$ deterministic algorithm for separating by a (convex) prism.
2. An $O(n^2 \log n)$ expected-time randomized algorithm for separating by a slab.
3. An $O(n^2 \log^2 n \log^2 (\log n))$ expected-time randomized or an $O(n^2 \text{polylog}(n))$ deterministic algorithm for separating by a (convex dihedral) wedge.
4. An $O(n^3 \log^2 n \log^2 (\log n))$ expected-time randomized or an $O(n^3 \text{polylog}(n))$ deterministic algorithm for separating by a double wedge.

Finally, we also study the problem of separating by a (convex) cone. Currently we do not have any subcubic algorithm for cone separability, but we prove a number of structural properties of cone separability that, we believe, should yield an $O(n^2 \text{polylog}(n))$ algorithm for this problem.

Our results rely on various techniques from the algorithmic theory of arrangements and on geometric data structures. These results improve the previously known bounds by at least an order of magnitude and, in the case of prisms, by two orders of magnitude. We believe that our approach can be extended to derive improved algorithms for other separability problems in $\mathbb{R}^3$, including ones for which no non-trivial solutions are currently known, such as when $\mathcal{F}$ is the family of cylinders or circular cones. We consider our results to be of particular appeal due to the conceptual simplicity of the studied problems.

2 General Approach

Before presenting our specific results, we devote this section to describing the general approach taken in this paper, as well as a few geometric concepts that will be crucial for our algorithms.

**Minimal separators and their representation.** We call a body $F \in \mathcal{F}$ a container (with respect to $\mathcal{R}$) if $\mathcal{R} \subseteq F$. A container is called minimal if there is no other container $G$ in $\mathcal{F}$ with $G \subset F$. Finally, a (minimal) container $F$ is called a (minimal) separator if $\mathcal{B} \cap \text{int}(F) = \emptyset$.

The families $\mathcal{F}$ that we study in this paper have the property that a minimal container can be represented as a point in a low-dimensional space despite the possibility that its combinatorial complexity may be $\Omega(n)$. For example, a minimal prism container can be represented by the direction of its axis. Indeed, if we fix a direction $u$ on the unit sphere $S^2$, then there is a unique minimal prism container, namely, the one formed by the Minkowski sum of $\text{conv}(\mathcal{R})$ with a line in direction $u$. Hence, the set of minimal prism containers can be represented by $S^2$. Similarly, if $\mathcal{F}$ is the family of cones in $\mathbb{R}^3$, then a minimal cone container can be represented by its apex, a point in $\mathbb{R}^3$.

Let $\mathcal{C} = \mathcal{C}(\mathcal{F}, \mathcal{R})$ be the parametric space that represents the set of minimal containers in $\mathcal{F}$, i.e., each point $F \in \mathcal{C}$ corresponds to a minimal container in $\mathcal{F}$ with respect to $\mathcal{R}$. With a slight abuse of notation, we will not distinguish between a minimal container $F \in \mathcal{F}$ and the corresponding point in $\mathcal{C}$. Given a set $\mathcal{B}$ of points in $\mathbb{R}^3$, let $\Sigma = \Sigma(\mathcal{B}, \mathcal{R}, \mathcal{F}) \subseteq \mathcal{C}$ denote the set of minimal separators. For a point $p \in \mathbb{R}^3$, let $\mathcal{K}_p \subseteq \mathcal{C}$ denote the set of minimal containers that also contain $p$, i.e., for each $C \in \mathcal{K}_p$, $\text{conv}(\mathcal{R}) \cup \{p\} \subseteq C$. Then $\Sigma = \mathcal{C} \setminus \bigcup_{p \in \mathcal{B}} \text{int}(\mathcal{K}_p)$. The problem of determining whether a body $F \in \mathcal{F}$ separates $\mathcal{B}$ from $\mathcal{R}$ is equivalent to determining whether $\Sigma \neq \emptyset$.

One possible approach is to bound the combinatorial complexity of $\Sigma$ and to compute a boundary representation of $\Sigma$ in time proportional to its complexity. However, in most cases we do not compute $\Sigma$ in its entirety due to its potentially large combinatorial complexity. Instead, we prove various geometric/topological properties of $\Sigma$ and design faster algorithms for detecting whether $\Sigma \neq \emptyset$. For example, we show that if $\mathcal{F}$ is the set of prisms in $\mathbb{R}^3$, then $\Sigma(\mathcal{B}, \mathcal{R}, \mathcal{F})$ can be represented as the intersection of two regions $X$ and $Y$. We compute a compact representation of each of $X$ and $Y$ and using this compact representation determine in $O(n \log^4 n)$ time whether $X \cap Y \neq \emptyset$. Similarly, for the case of wedges and double wedges, we search over various cross-sections of $\Sigma$ to detect its non-emptiness.

We make the following assumptions for the sake of simplicity. The algorithms can easily be modified to work for the general case.

1. The points $\mathcal{R} \cup \mathcal{B}$ are in general position, in the sense that no two points have the same $x$, $y$- or $z$-coordinate, no two points lie on a line passing through the origin, no three points are collinear or lie in a common vertical plane, and no four points are co-planar. We also assume that the origin $o$ lies in the interior of $\text{conv}(\mathcal{R})$.
2. $\Sigma = \text{cl}(\mathcal{C} \setminus \bigcup_{p \in \mathcal{B}} \text{conv}(\mathcal{K}_p))$.
3. Since $\mathcal{F}$ is a family of convex bodies in Sections 3–5 (note that double wedges are not convex), we assume that $\text{int}(\text{conv}(\mathcal{R})) \cap \mathcal{B} = \emptyset$ in these sections. Indeed if $\text{int}(\text{conv}(\mathcal{R})) \cap \mathcal{B} \neq \emptyset$, then no body in $\mathcal{F}$ can separate $\mathcal{R}$ from $\mathcal{B}$ and, since we are
dealing with point sets in 3-space, we can check this condition in \( O(n \log n) \) time.

**Arrangements.** Let \( \mathcal{H} \) be a set of hyperplanes in \( \mathbb{R}^d \). The arrangement of \( \mathcal{H} \), denoted by \( \mathcal{A}(\mathcal{H}) \), is the decomposition of \( \mathbb{R}^d \) into (relatively open) cells so that each cell is a maximal connected region that lies in the same subset of \( \mathcal{H} \). Each cell of \( \mathcal{A}(\mathcal{H}) \) is a convex polyhedron. The combinatorial complexity of a \( k \)-dimensional cell \( C \), denoted as \( |C| \), is the number of cells of \( \mathcal{A}(\mathcal{H}) \) of dimension less than \( k \) that are contained in \( \partial C \), the boundary of \( C \); see [1] for a recent survey on arrangements. It is well known that the complexity of a single cell in \( \mathcal{A}(\mathcal{H}) \) is \( O(n^{d/2}) \) and that the complexity of the entire arrangement is \( O(n^d) \). The lower (resp. upper) envelope of \( \mathcal{H} \) is the boundary of the \( d \)-dimensional cell of \( \mathcal{A}(\mathcal{H}) \) that lies below (resp. above) all hyperplanes of \( \mathcal{H} \). The zone of \( \mathcal{H} \) with respect to a surface \( \gamma \), denoted as \( Z(\mathcal{H}, \gamma) \), is the set of \( d \)-dimensional cells of \( \mathcal{A}(\mathcal{H}) \) that intersect \( \gamma \). A cell \( C \in Z(\mathcal{H}, \gamma) \) is \( \mathcal{A}(\mathcal{H}) \)-tight if \( \gamma \) is a convex surface and \( O(n^{d-1}) \) if \( \gamma \) is a hyperplane [4].

**3 Prism Separability**

In this section we assume \( \mathcal{F} \) to be the set of (convex) prisms in \( \mathbb{R}^3 \). Recall that a prism \( F \) is the Minkowski sum of a closed, convex body \( C \) with a line \( \ell \). The direction of \( \ell \) is referred to as the direction of \( F \). Since \( \mathcal{F} \) is a family of convex bodies, we describe the algorithm under Assumption (A3). As mentioned in Section 2, for a direction \( u \in \mathbb{S}^2 \), the Minkowski sum of \( \text{conv}(\mathcal{R}) \) and a line in \( u \) is minimal if \( \text{conv}(\mathcal{R}) \) and \( u \) are the same, but for simplicity we distinguish them. We thus represent \( \mathcal{C} \), the set of minimal prism containers, by \( \mathbb{S}^2 \).

We first analyze the structure of \( \Sigma \) and then describe the algorithms for determining whether \( \Sigma \neq \emptyset \).

**Structure of \( \Sigma \).** For a point \( p \in \mathbb{R}^3 \setminus \text{conv}(\mathcal{R}) \), let \( C_p \) be the union of rays emanating from \( p \) and intersecting \( \text{conv}(\mathcal{R}) \), i.e., \( C_p \) is the cone spanned by \( p \) and \( \text{conv}(\mathcal{R}) \).

Set \( S_p = (C_p - p) \cap \mathbb{S}^2 \), i.e., the set of directions \( u \) such that the ray emanating from \( p \) in direction \( u \) intersects \( \text{conv}(\mathcal{R}) \); here \( C_p - p \) denotes the cone \( C_p \) translated by \( p \), i.e., such that its apex lies at the origin. The following lemmas are straightforward.

**Lemma 3.1.** Let \( p \) and \( q \) be two points in \( \mathbb{R}^3 \). If \( p \in C_q \), then \( S_q \subseteq S_p \) (see Figure 1(i)).

**Lemma 3.2.** For any point \( p \in \mathbb{R}^3 \), \( \mathcal{K}_p = S_p \cup (-S_p) \).

Hence,\[ \Sigma = \mathbb{S}^2 \setminus \bigcup_{b \in \mathcal{B}} \text{int} (S_b \cup -S_b) = \text{cl} (\mathbb{S}^2 \setminus \bigcup_{b \in \mathcal{B}} (S_b \cup -S_b)) . \]

The second equality follows from Assumption (A2). Let \( \mathcal{K} = \bigcup_{b \in \mathcal{B}} S_b \). Then \[ \Sigma = \text{cl}(\mathbb{S}^2 \setminus (\mathcal{K} \cup -\mathcal{K})) . \]

Hurtado et al. [15] have proved that the complexity of \( \Sigma \) is \( O(n^2 \alpha(n)) \). Here we present a topological property of \( \Sigma \) that will be useful in developing the algorithm for detecting the emptiness of \( \Sigma \).

Consider a collection \( \mathcal{S} \) of closed, simply connected regions on a topological disk or a sphere. We say that \( A, B \in \mathcal{S} \) cross if both \( A \setminus B \) and \( B \setminus A \) are disconnected; see Figure 1(ii). \( \mathcal{S} \) is considered a family of pseudo-disks if no two of its members cross. Let \( \mathcal{S} = \{ S_b \mid b \in \mathcal{B} \} \). For simplicity, we assume that \( \partial S_p \) and \( \partial S_q \), for any \( p \neq q \in \mathcal{B} \), intersect transversally if at all.

![Figure 1](image-url)

**Figure 1:** (i) \( p \in C_q \), so \( S_q \subseteq S_p \); (ii) \( S_p \) and \( S_q \) cross.

**Lemma 3.3.** \( \mathcal{S} \) is a family of pseudo-disks.

**Proof.** It is sufficient to argue, for \( p, q \in \mathcal{B} \), that \( S_p \) and \( S_q \) do not cross. Suppose \( S_p \) and \( S_q \) cross and thus their boundaries intersect in at least four points; see Figure 1(ii). (Since \( S_p \) and \( S_q \) are closed curves, they cannot intersect three times.) Then there are at least four planes, say, \( \pi_1, \ldots, \pi_4 \), through the origin that are common tangents to the cones \( C_p - p \) and \( C_q - q \). (Each of the two cones has its apex at the origin.)

We claim that, for each \( \pi_i \), there exists a plane \( \pi'_i \) parallel to \( \pi_i \), passing through \( p \) and \( q \), and tangent to \( C_p \) and \( C_q \). Indeed consider two planes parallel to \( \pi_i \): \( \pi'_i \) passing through \( p \) and \( \pi''_i \) passing through \( q \). Since the oriented plane \( \pi_i \) passes through the origin and is tangent to \( C_p - p, \pi'_i \) is tangent to \( C_p \) and thus to \( \text{conv}(\mathcal{R}) \). Similarly, \( \pi''_i \) must be tangent to \( \text{conv}(\mathcal{R}) \) as well. This is impossible, unless \( \pi'_i \) and \( \pi''_i \) are the same oriented plane, completing the proof of the claim.

We thus have four planes, each passing through the line \( pq \) and tangent to \( \text{conv}(\mathcal{R}) \), but there are at most two such planes, a contradiction. Hence, \( \mathcal{S} \) is a family of pseudo-disks. \( \square \)
**The algorithm.** We first describe a simple near-quadratic algorithm for computing $\Sigma$, and then we describe an $O(n \log^4 n)$ algorithm that determines whether $\Sigma = \emptyset$.

Compute $S_b$, for each $b \in B$, identifying the facets of $C_b$ (or equivalently the facets of $\text{conv}(R \cup \{b\})$ incident upon $b$) by a brute-force traversal of $\partial(\text{conv}(R))$. This yields, in $O(n^2)$ time, the family $S = \{S_b \mid b \in B\}$ of $n$ convex spherical polygons of at most $n$ edges each, for a total of $O(n^2)$ edges. Since $S$ is a family of pseudodisks, any two boundaries cross at most twice, and the arrangement $A(S)$ has at most $2\binom{n}{2} = O(n^2)$ boundary intersection vertices. Thus it has total complexity $O(n^2)$ and can be computed in $O(n^2 \log n)$ time by a standard procedure [1]. The union $K = \bigcup S_b$ can be extracted from the arrangement in $O(n^2)$ time by a single traversal. Next, we can compute $-K$ in additional $O(n^2)$ time. Finally, we compute $K \cup -K$ by identifying the intersection points of $\partial K$ and $-\partial K$ and then traversing the two regions. Since $K \cup -K$ has $O(n^2 \alpha(n))$ vertices, the total time spent in this step is $O(n^2 \alpha(n) \log n)$. Finally, we set $\Sigma = cl(S^2 \setminus (K \cup -K))$. Thus we have proven the following.

**Theorem 3.1.** Given two sets of points $R$ and $B$ in $\mathbb{R}^d$, each of cardinality $n$, we can compute the set of all minimal prism separators in $O(n^2 \alpha(n) \log n)$ time.

Next, we show that we can detect the emptiness of $\Sigma$ in $O(n \log^4 n)$ time by computing an implicit representation of $K$ and deciding whether $S^2 = K \cup -K$. Using the Dobkin-Kirkpatrick hierarchy of convex polyhedra [9] on $\text{conv}(R)$ and noting that it can be viewed as an implicit hierarchical representation of $C_p$, for any $p \in \mathbb{R}^d$ (see [9]), we have the following tool needed in the algorithms below.

**Lemma 3.4.** We can preprocess $\text{conv}(R)$ in $O(n \log n)$ time so that the following operations can be performed in $O(\log n)$ time.

(i) Given a direction $u \in S^2$ and a point $p \in \mathbb{R}^d$, determine whether $u \in S_p$.

(ii) Given a direction $u \in S^2$ and a point $p \in \mathbb{R}^d$, determine the at most two great circles passing through $u$ and tangent to $S_p$.

Note that (i) is equivalent to testing whether the line through $p$ in direction $u$ meets $\text{conv}(R)$ and (ii) is equivalent to computing the planes through $p$ tangent to $\text{conv}(R)$ and parallel to $u$. Also observe that for distinct point $p, q \in \mathbb{R}^d$, $p \in C_p$ or $q \in C_q$, if and only if line $pq$ meets $\text{conv}(R)$, which again can be checked in $O(\log n)$ time. Using the above lemma, we prove the following.

**Lemma 3.5.** We can preprocess $\text{conv}(R)$ in $O(n \log n)$ time so that for any two points $p, q \in \mathbb{R}^3$, $p \notin C_q$ and $q \notin C_p$, we can determine in $O(\log^2 n)$ time the intersection points of $\partial S_p$ and $\partial S_q$, if they exist.

**Proof.** We construct in $O(n \log n)$ time the Dobkin-Kirkpatrick hierarchy on $\text{conv}(R)$. Note that $S_p$ and $S_q$ are disjoint if and only if $C_p - p$ and $C_q - q$ are disjoint. Using the hierarchy we can determine in $O(\log^2 n)$ time whether $C_p - p$ and $C_q - q$ are disjoint. Henceforth we assume that they intersect.

Let $\pi_{pq}$ be the plane passing through $p$, $q$, and the origin $o$. Let $\gamma_p = S_p \cap \pi_{pq} = (C_p - p) \cap \pi_{pq} \cap S^2$; $\gamma_p$ is a (great) circular arc. Let $l_p$ and $r_p$ be the endpoints of $\gamma_p$, $l_p, r_p \in \partial S_p$. Similarly, define $\gamma_q = S_q \cap \pi_{pq}$, and let $l_q, r_q$ be its endpoints; $l_q, r_q \in \partial S_q$. There are three cases:

(i) $\gamma_p \subseteq \gamma_q$: In this case, $q \in C_p$ and, by Lemma 3.1, $S_p \subseteq S_q$ and we do not have to compute intersections between the boundaries of $S_p$ and $S_q$. See Figure 2(i).

(ii) $\gamma_p$ and $\gamma_q$ overlap, but they are not nested: Suppose $l_p, l_q, r_p, r_q$ appear in this order along the great circle $\pi_{pq} \cap S^2$; see Figure 2(ii). Then we have $l_p \notin S_q, r_p \in S_q$, $l_q \in S_p$, and $r_q \notin S_p$. Therefore these four points partition each of $\partial S_p$ and $\partial S_q$ into two convex arcs $\partial S_p^+, \partial S_p^-$ and $\partial S_q^+, \partial S_q^-$ so that $\partial S_p^+, \partial S_q^-$ intersect at one point and $\partial S_p^-, \partial S_q^+$ intersect at one point. By traversing a path in the Dobkin-Kirkpatrick hierarchy on $\text{conv}(R)$ and using Lemma 3.4 at each node on the path to determine which child of that node should be visited next, we can determine the intersection points of $\partial S_p$ and $\partial S_q$ in $O(\log^2 n)$ time.

(iii) $\gamma_p$ and $\gamma_q$ are disjoint: Let $u_p \in S^2$ (resp. $u_q$) be the direction of the ray $\overrightarrow{p0}$ (resp. $\overrightarrow{q0}$). Then $u_p \in S_q \setminus S_q$ and $u_q \in S_q \setminus S_p$. Let $p\rho$ be the ray $p - t u_p$, $t \geq 0$. As we move a point $x$ on $p\rho$ starting from $p$, $x$ shrinks continuously toward the point $u_p$. Let $w \in p\rho$ be the point such that $S_q$ and $S_w$ are tangent to each other (i.e., the maximum value of $t$ at which $S_q$ and $S_{p(t-u_p)}$ intersect). Let $v = \partial S_w \cap \partial S_q$. The points $w$ and $v$ can be computed in $O(\log^2 n)$ time using Lemma 3.4. Similarly, we move a point $y$ along the ray $q - tu_q$ until $S_p$ and $S_q$ become tangent to each other at a point $v_q$. Let $z_q$ (resp. $z_p$) be the other intersection point of $\partial S_q$ (resp. $\partial S_q$) with the great circle passing through $v_q$ and $v_q$. By construction, $z_p \notin S_q$ and $z_q \notin S_q$. After having computed $v_p, v_q, z_p$, and $z_q$, we can compute the intersection points of $\partial S_p$ and $\partial S_q$ in $O(\log^2 n)$ time, since these four points partition $\partial S_p$ and $\partial S_q$ as in case (ii) above. □

We now describe how to detect whether $\Sigma$ is empty. To simplify the presentation, we view $S$ as a set of planar
convex polygons. For any subset $X \subseteq S$, we represent each connected component of $\partial (\bigcup X)$ as a sequence of maximal $x$-monotone arcs, each lying on the top or the bottom boundary of a single polygon $S_b \in X$. For each arc $\gamma$, we store: (i) the coordinates of its endpoints, (ii) the point $b \in B$ such that $\gamma \subseteq \partial S_b$, and (iii) a bit specifying whether $\gamma$ lies on the top or the bottom boundary of $\partial S_b$; we refer to $\gamma$ as a top arc in the former case and as a bottom arc in the latter case. Let $X^*$ denote the set of arcs in the implicit representation of $X$. The endpoints of arcs in $X^*$ are the leftmost and the rightmost points of polygons in $X$ and the intersection points between the boundaries of two polygons of $X$ that appear on $\partial (\bigcup X)$. Since there are $O(|X|)$ such points (cf. Lemma 3.3 and [17]), our implicit representation requires $O(n)$ space. Suppose we have such an implicit representation $X^*, Y^*$ for two subsets $X, Y \subseteq S$. Then we can compute the implicit representation $(X \cup Y)^*$ of the union of polygons in $X \cup Y$ by a sweep-line algorithm, as shown in [17]. The primitive step in this procedure is computing the intersection points between two arcs $\gamma \in X^*$ and $\gamma' \in Y^*$. Using Lemma 3.5, we can compute these intersection points in time $O(\log^3 n)$, so the sweep-line algorithm takes $O((|X| + |Y|) \log^3 n)$ time. Hence, using a divide-and-conquer technique, we can compute an implicit representation of $K$ in $O(n \log^4 n)$ time. After having computed $K^*$, we can compute the implicit representation $(-K)^*$ of $-K$ in linear time.

We would like to perform another sweep to compute an implicit representation of $K \cup (-K)$. But, unfortunately, $\partial S_p$ and $-\partial S_q$ can intersect as many as a linear number of times, so we cannot compute the same implicit representation of $\Sigma$. Instead, we detect the emptiness of $\Sigma$ using the following observation. If $\Sigma = c(S^2 \setminus (K \cup (-K))) \neq \emptyset$, then the leftmost point of a connected component of $\Sigma$ is either an endpoint of an arc in $K^* \cup (-K)^*$ or an intersection point of a top arc of $K^*$ (resp. $(-K)^*$) with a bottom arc of $(-K)^*$ (resp. $K^*$). We first check in a total of $O(n \log^2 n)$ time, by a sweep-line algorithm, whether an endpoint of any arc in $K^*$ lies outside $-K$, or vice versa. Next, we observe that we can compute intersection points of the top boundary of $S_p$ with the bottom boundary of $S_q$, for $p, q \in B$, in $O(\log^2 n)$ time using Lemma 3.4. Hence, using a sweep-line algorithm, we can determine in $O(n \log^2 n)$ time whether any top arc of $K^*$ intersects a bottom arc of $(-K)^*$, or vice versa. Putting everything together, we obtain the following.

**Theorem 3.2.** Given two sets of points $R$ and $B$ in $\mathbb{R}^3$, each of cardinality $n$, we can determine in $O(n \log^4 n)$ time the existence of a prism that contains $R$ but whose interior is disjoint from $B$ and return such a prism if it exists.

### 4 Slab Separability

We now let $F$ be the family of all slabs, i.e., closed regions delimited by two parallel planes. For a given direction $u \in S^2$, there is a unique minimal slab container $\sigma(u)$: the planes bounding $\sigma(u)$ are normal to $u$ and support $\text{conv}(R)$; one of these two planes lies above $\text{conv}(R)$ and the other lies below $\text{conv}(R)$. Vertical slabs can be handled separately—use a two-dimensional algorithm for separating planar point sets (x-y-projections of $R$ and $B$) by a strip. If $\sigma(u)$ is a separator, then none of the points in $B$ lie in the interior of $\sigma(u)$. Using a standard duality transform, we can map $R$ (resp. $B$) to a set of planes $R^*$ (resp. $B^*$). The dual
$\sigma^*$ of a minimal separating slab is a vertical segment one of whose endpoints lies on the lower envelope $\mathcal{L}$ of $\mathcal{A}(R^*)$ and the other on the upper envelope $\mathcal{U}$ of $\mathcal{A}(R^*)$. Moreover $\sigma^*$ lies completely inside a cell of $\mathcal{A}(B^*)$ as it does not intersect any plane of $B^*$. See Figure 3. Based on these observations, we proceed as follows.

![Diagram](image)

**Figure 3:** Separating by a slab: (i) A slab separating $R$ from $B$ in the primal setting. (ii) Dual setting: solid lines form $R^*$, dashed lines form $B^*$, vertical segment is $\sigma^*$, and the shaded region is the cell of $\mathcal{A}(B^*)$ that contains the vertical segment $\sigma^*$.

Let $\mathcal{Z}$ be the set of cells in $\mathcal{A}(B^*)$ that intersect both $\mathcal{L}$ and $\mathcal{U}$. Since each cell $C \in \mathcal{Z}$ intersects the convex surface $\mathcal{L}$, $\sum_{C \in \mathcal{Z}} |C| = O(n^2 \log n)$ [4]. For a cell $C \in \mathcal{Z}$, let $C^+ = C \cap \mathcal{U}$ and $C^- = C \cap \mathcal{L}$.

**Lemma 4.1.** $\sum_{C \in \mathcal{Z}} (|C^+| + |C^-|) = O(n^2)$.

**Proof.** Each vertex of $\partial C^+$ is an intersection point of an edge of $\mathcal{U}$ and a plane of $B^*$, or a face of $\mathcal{U}$ and a line formed by the intersection of two planes of $B^*$. Since $\mathcal{U}$ has $O(n)$ edges, there are $O(n^2)$ vertices of the first type. As an intersection line of two planes of $B^*$ intersects the convex surface $\mathcal{U}$ at most twice, there are $O(n^2)$ intersections of the second type as well. Regions $C^+$ are treated symmetrically.

We compute $\mathcal{Z}$ as well as $C^+, C^-$, for each cell $C \in \mathcal{Z}$, in $O(n^2 \log n)$ expected time, using a lazy randomized incremental algorithm [8]. We omit the details from this abstract. For each cell $C \in \mathcal{Z}$, we project $C^+$ and $C^-$ onto the $xy$-plane, yielding two families of disjoint convex polygons. Using a sweep-line algorithm, we determine in $O((|C^+| + |C^-|) \log n)$ time whether the $xy$-projections of $C^+$ and $C^-$ intersect. If they intersect at a point $\xi$, then the slab dual to the vertical segment $\mathcal{U}(\xi) \cap \mathcal{L}(\xi)$ is a minimal separating slab, where $\mathcal{U}(\xi)$ (resp. $\mathcal{L}(\xi)$) is the intersection point of $\mathcal{U}$ (resp. $\mathcal{L}$) with the vertical line through $\xi$. Repeating this step for all cells of $\mathcal{Z}$, we can determine whether there exists a (minimal) separating slab. By Lemma 4.1, we obtain the following:

**Theorem 4.1.** Given two sets of points $R$ and $B$ in $\mathbb{R}^3$, each of cardinality $n$, an algorithm can determine the existence of a separating slab and return such a slab if it exists in time $O(n^2 \log n)$.

## 5 Wedge Separability

We now let $\mathcal{F}$ be the family of wedges, i.e., intersections of two closed halfspaces. If the two bounding planes are parallel, then the resulting wedge is a slab. Since we have already studied slab-separability, we assume that the wedges in $\mathcal{F}$ are bounded by non-parallel planes. As in the case of slabs, the two boundary planes of a minimal separating wedge support $\mathcal{U}(R)$; see Figure 4(i). For a direction $u \in \mathbb{S}^2$, let $\pi(u)$ denote the (oriented) plane with outward normal $u$ that supports $\mathcal{U}(R)$, and let $B(u) \subseteq B$ denote the subset of points that lie on the same side of $\pi(u)$ as $R$. A minimal wedge container can be represented by a pair of outward normals $(u_1, u_2) \in \mathbb{S}^2 \times \mathbb{S}^2$. If $(u_1, u_2)$ is a separator, then $\pi(u_2)$ separates $\mathcal{U}(R)$ and $\mathcal{U}(B(u_1))$. As we vary the direction $u$, $B(u)$ remains the same until $\pi(u)$ passes through a point of $B$. We proceed as follows to compute a minimal separating wedge, if one exists.

Let $\mathcal{N}$ be the normal diagram (also known as the Gauss map) of $\mathcal{U}(R)$, i.e., the subdivision of $\mathbb{S}^2$ into maximal regions so that for all directions $u$ within each region, the plane $\pi(u)$ is tangent to the same feature (vertex, edge, face) of $\mathcal{U}(R)$. $\mathcal{N}$ can be computed in linear time from $\mathcal{U}(R)$. For a point $b \in B$, let $\gamma_b \subset \mathbb{S}^2$ be the locus of directions $u$ so that $\pi(u)$ passes through $b$; if $b \notin \partial \mathcal{U}(R)$, $\gamma_b$ is the boundary of a convex polygon on $\mathbb{S}^2$, namely the set of outer normals to the planes tangent to $C_b$ (defined in Section 3). The curve $\gamma_b$ consists of arcs of great circles, the arc ends at points lying on the edges of $\mathcal{N}$. We compute the arrangement of $\Gamma = \{ \gamma_b \mid b \in B \}$. The following lemma follows from Lemma 3.3 and the first algorithm described in Section 3.

**Lemma 5.1.** $\mathcal{A}(\Gamma)$ has $O(n^2)$ vertices, including the breakpoints of arcs in $\Gamma$, and it can be computed in $O(n^2 \log n)$ time.

By construction and the above discussion, for all directions $u$ within the same face $\phi$ of $\mathcal{A}(\Gamma)$, $B(u)$ is the same, which we denote by $B(\phi)$. Moreover, if $\phi, \phi'$ are two adjacent faces of $\mathcal{A}(\Gamma)$, then $|B(\phi) \oplus B(\phi')| \leq 1$, where $\oplus$ denotes symmetric difference. We wish to determine whether there is a face $\phi \in \mathcal{A}(\Gamma)$ for which $\mathcal{U}(R)$ and $\mathcal{U}(B(\phi))$ are disjoint (i.e., weakly linearly separable). Roughly speaking, we traverse the faces of $\mathcal{A}(\Gamma)$, updating the set $B(\phi)$ as we go from one face to another, and use a dynamic data structure.
to determine whether \( \text{conv}(\mathcal{R}) \) and \( \text{conv}(\mathcal{B}(\phi)) \) ever become disjoint.

In more detail, we compute an Eulerian tour of the dual graph of \( \mathcal{A}(\Gamma) \), which is a sequence \( \Phi = \langle \phi_1, \phi_2, \ldots, \phi_k \rangle \), \( k = O(n^2) \), of faces of \( \mathcal{A}(\Gamma) \). We traverse the sequence \( \Phi \), and at each \( i \), we maintain \( \text{conv}(\mathcal{R}) \cap \text{conv}(\mathcal{B}(\phi_i)) \) in a dynamic data structure. Since we know the sequence of insertions and deletions in advance and only want to determine whether \( \text{conv}(\mathcal{R}) \cap \text{conv}(\mathcal{B}(\phi_i)) \) ever becomes empty, we can use the off-line data structure by Epstein [11], which can perform an update in amortized expected time \( O(\log^2 n \log^2 (\log n)) \). He also described another data structure that can perform an update in polylog \( n \) amortized deterministic time. We conclude the following.

**Theorem 5.1.** Given two sets of points \( \mathcal{R} \) and \( \mathcal{B} \) in \( \mathbb{R}^3 \), each of cardinality \( n \), an algorithm can determine the existence of a separating convex dihedral wedge, and return such a wedge if it exists, in randomized expected time \( O(n^2 \log n \log^2 (\log n)) \) or in \( O(n^2 \text{polylog}(n)) \) worst-case time.

6 Double-Wedge Separability

In this section, \( \mathcal{F} \) is the family of all double wedges, i.e., (closures of) a symmetric difference of two halfspaces bounded by non-parallel planes. Without loss of generality we assume that we consider only those double wedges that do not contain a vertical plane.

Once again, consider the dual version of the problem, where we are required to determine, given two collections of planes \( \mathcal{R}^*, \mathcal{B}^* \) whether there exists a segment \( \sigma^* \) that intersects all of \( \mathcal{R}^* \) but none of \( \mathcal{B}^* \). Let \( C \) be a 3-dimensional cell of \( \mathcal{A}(\mathcal{R}^* \cup \mathcal{B}^*) \). For a plane \( h \in \mathcal{R}^* \cup \mathcal{B}^* \), let \( h^{\text{int}}_{\mathcal{R}^*} \) (resp. \( h^{\text{int}}_{\mathcal{B}^*} \)) be the closed halfspace bounded by \( h \) that contains (resp. does not contain) \( C \). Let \( \mathcal{B}^*_C = \{ h^{\text{int}}_{B} | h \in \mathcal{B}^* \} \), and let \( \mathcal{R}^{\text{ext}}_C = \{ h^{\text{ext}}_{R} | h \in \mathcal{R}^* \} \).

If \( \sigma^* \) is the dual of a separating double wedge and one endpoint \( \xi \) of \( \sigma^* \) lies in \( C \), then the other endpoint \( \zeta \) of \( \sigma^* \) has to lie in \( P_C = \bigcap g \), where the intersection is taken over all halfspaces \( g \in \mathcal{B}^{\text{ext}} \cup \mathcal{R}^{\text{ext}}_C \). Indeed, \( \xi \) and \( \zeta \) lie on the same side of all planes in \( \mathcal{B}^* \) and on the opposite sides of all planes in \( \mathcal{R}^* \). Hence the segment \( (\xi, \zeta) \) does not intersect any plane of \( \mathcal{B}^* \) and intersects all planes of \( \mathcal{R}^* \).

As in the previous section, consider a dynamic data structure that can maintain \( P_C \) through insertions and deletions in \( \mathcal{B}^{\text{ext}} \cup \mathcal{R}^{\text{ext}}_C \). Traverse the cells \( C \) of \( \mathcal{A}(\mathcal{R}^* \cup \mathcal{B}^*) \) and maintain \( P_C \) in this fashion. If for some cell \( C \) during the traversal, \( P_C \) is nonempty we return a segment \( \sigma^* = (\xi, \zeta) \) such that \( \xi \in C \) and \( \zeta \in P_C \), otherwise we conclude that a segment \( \sigma^* \) as desired does not exist.

The running time in the theorem below follows from the analysis in the preceding section.

**Theorem 6.1.** Given two sets of points \( \mathcal{R} \) and \( \mathcal{B} \) in \( \mathbb{R}^3 \), each of cardinality \( n \), an algorithm can determine the existence of a separating dihedral double wedge and return such a double wedge if it exists in randomized expected time \( O(n^3 \log^2 n \log^2 (\log n)) \) or \( O(n^2 \text{polylog}(n)) \) worst-case time.

7 Cone Separability

In this section \( \mathcal{F} \) is the family of all (closed, convex, infinite) cones. Namely, given a point \( p \in \mathbb{R}^3 \) and a closed, convex body \( K \setminus \{p\} \), a cone \( C(p, K) \) is the union of rays emanating from \( p \) that intersect \( K \); \( p \) is the apex of \( C(p, K) \). If \( p \in K \), then \( C(p, K) = \text{aff}(K) \), and if \( p \) is at infinity then \( C(p, K) \) is a prism. In this section, we will assume for simplicity that the apex of the container cone is always at a point of \( \mathbb{R}^3 \) strictly outside of \( \text{conv}(\mathcal{R}) \). For any point \( p \in \mathbb{R}^3 \), there is a unique minimal cone container (with respect to \( \mathcal{R} \)) with apex \( p \), namely the cone \( C_p := C(p, \text{conv}(\mathcal{R})) \), so we represent the set \( C \) of minimal cone containers by points in \( \mathbb{R}^3 \setminus \text{conv}(\mathcal{R}) \).

Unlike the cases considered in previous sections,
here we only discuss the structure of the space $\Sigma$ of minimal cone separators. This investigation yields an $O(n^2)$ bound on the combinatorial complexity of $\Sigma$. We do not at this point have a near-quadratic algorithm for constructing $\Sigma$, or an efficient way of testing whether it is empty. The fastest known algorithm for computing it is roughly cubic [16]; see the discussion below.

**Structure of $\Sigma$.** Fix a point $p \in B$ and examine the set $K_p := \{x \mid p \in C_x\}$. Let the *outer cone* $O_p$ of $p$ be the cone with apex $p$ and antipodal to $C_p$. Let the *inner cone* $I_p$ of $p$ be $C_p$ less the connected component of $C_p \setminus \text{conv}(R)$ containing $p$. See Figure 5 and note that we have slightly abused the notation in that the inner cone is not exactly a cone due to the presence of a “cap”, a portion of $\partial \text{conv}(R)$, on its boundary.) It is easy to see that $K_p = O_p \cup I_p$. In particular, putting $O := \bigcup O_p$ and $I := \bigcup I_p$, where the union is taken over $p \in B$, $\Sigma$ is the complement of $O \cup I$ and we focus on this latter set.

We call a ray $r$ *outgoing* if it emanates from a point $x$ in $\text{conv}(R)$ and *strictly outgoing* if $x$ lies in the interior of $\text{conv}(R)$. The following lemmas, presented here without proofs, form the heart of our analysis of geometric structure of $I$, $O$, and $\Sigma$. To avoid excessive notation, we think of a ray as being traversed from its origin to infinity. A tail of a ray $r$ is any ray $r' \subseteq r$.

**Lemma 7.1.** The following properties of outgoing rays and inner and outer cones hold:

1. If an outgoing ray meets an outer cone, it stays in it.
2. If a strictly outgoing ray meets an outer cone, it enters its interior and stays in it.
3. Every outgoing ray starts in each inner cone and then leaves, never to re-enter (trivially, by convexity of the inner cone).

4. The boundary of an outer cone is a union of tails of outgoing rays, disjoint except at its apex.
5. The boundary of an inner cone, except for its cap (i.e., $(\partial I_p) \setminus \text{conv}(R)$), is a disjoint union of outgoing rays.

Intuitively, the above properties allow us to treat $\partial O$ as the lower envelope of $\{\partial O_p \mid p \in B\}$, since any outgoing ray, having encountered any point of an outer cone never leaves it again. Similarly, and perhaps counterintuitively, $\partial I$ acts as an upper envelope. Finally, if we were to take the approach of [5] to computing these unions by examining, for each point $p$, $B_p := \partial O \cap \partial O_p$, the last two properties allow us to look at the boundary of each inner or outer cone as a surface with a natural polar-like coordinate system—one coordinate is the choice of the outgoing ray on the boundary and the second is the distance from $p$ along this ray. We observe that $B_p$ is in fact the region below the lower envelope of the sets $D_{p,q} := \partial O_p \cap \partial O_q$, $q \in B \setminus \{p\}$. Another useful fact we prove is

**Lemma 7.2.** $\{D_{p,q} \mid q \in B \setminus \{p\}\}$ is a family of pseudodisks on $\partial O_p$, for any $p \in B$.

In particular, we can argue that the total complexity of all $B_p$ and thus of $\partial O$ is in fact $O(n^2)$. A similar argument applies to $\partial I$. In fact, a more careful counting implies that $\partial \Sigma$ restricted to $O_p$ (or $I_p$) is the “sandwich region” between two envelopes and can be argued by similar methods to have complexity $O(n^2)$.

**Theorem 7.1.** The complexity of the space $\Sigma$ of all minimal separator cones is $O(n^2)$.

**The (lack of an efficient) algorithm.** As already mentioned, we do not have a near-quadratic algorithm for computing $\Sigma$. A near-cubic one can be obtained by computing the inner and outer cones explicitly in roughly quadratic time. This yields a collection of $n$ convex polyhedra each of complexity $O(n)$. As observed in [16], their union can now be computed by the algorithm of [5] in roughly cubic randomized expected time. Any attempt to tune the union-of-polyhedra algorithm of [5] to this case that represents the cones explicitly is doomed to failure, as it starts by computing the intersection of every face of every polyhedron with the remaining polyhedra and the total complexity of all such intersections can be cubic in the worst case. A natural attempt to circumvent this is to represent the intersections implicitly, as was done in Section 3. Indeed, this is possible and a near-quadratic algorithm would result if one could compute, in say, polylogarithmic time, the at most two (by Lemma 7.2) points of intersection of any three cones.
References


