# Line Intersection Searching Amid Unit Balls in 3-Space

**Pankaj K. Agarwal**
Department of Computer Science, Duke University, Durham, NC, USA

**Esther Ezra**
School of Computer Science, Bar Ilan University, Ramat Gan, Israel

## Abstract

Let $\mathscr{B}$ be a set of $n$ unit balls in $\mathbb{R}^3$. We present a linear-size data structure for storing $\mathscr{B}$ that can determine in $O^*(n^{1/2})$ time whether a query line intersects any ball of $\mathscr{B}$ and report all $k$ such balls in additional $O(k)$ time. The data structure can be constructed in $O(n \log n)$ time. (The $O^*(\cdot)$ notation hides subpolynomial factors, e.g., of the form $O(n^\varepsilon)$, for arbitrarily small $\varepsilon > 0$, and their coefficients which depend on $\varepsilon$.)

We also consider the dual problem: Let $\mathscr{L}$ be a set of $n$ lines in $\mathbb{R}^3$. We preprocess $\mathscr{L}$, in $O^*(n^2)$ time, into a data structure of size $O^*(n^2)$ that can determine in $O^*(1)$ time whether a query unit ball intersects any line of $\mathscr{L}$, or report all $k$ such lines in additional $O(k)$ time.

**2012 ACM Subject Classification** Theory of computation; Theory of computation → Computational geometry

**Keywords and phrases** Intersection searching, cylindrical range searching, partition trees, union of cylinders

**Digital Object Identifier** 10.4230/LIPIcs.SoCG.2023.5

**Funding** *Pankaj K. Agarwal*: Work partially supported by NSF grants IIS-18-14493, CCF-20-07556, and CCF-22-23870.
*Esther Ezra*: Work partially supported by Grants 824/17 and 800/22 from the Israel Science Foundation.

## 1 Introduction

Let $\mathscr{B} := \{B_1, \ldots, B_n\}$ be a set of $n$ unit-radius balls in $\mathbb{R}^3$. We wish to preprocess $\mathscr{B}$ into a data structure that supports various *line-intersection* queries. That is, given a query line $\ell$ in $\mathbb{R}^3$, determine whether $\ell$ intersects a ball in $\mathscr{B}$, report all balls of $\mathscr{B}$ that $\ell$ intersects, count the number of such balls, or compute some aggregate function on the balls intersected by $\ell$. Since all balls in $\mathscr{B}$ have the same radius, this problem can be reformulated as the *unit-cylinder range-searching* problem: Consider the set $P$ of the centers of balls in $\mathscr{B}$. Preprocess $P$ into a data structure so that we can quickly answer range queries for a query unit-radius cylinder $C$, such as determine whether $C \cap P = \emptyset$ (referred to as *emptiness query*), report $C \cap P$ (*reporting query*), or compute $|C \cap P|$ (*counting query*). We also consider the dual problem where the input is a set $\mathscr{L}$ of $n$ lines in $\mathbb{R}^3$, and we wish to answer *unit-ball-intersection* queries, i.e., does a query unit ball intersect any line of $\mathscr{L}$.

**Related work** The intersection-searching problem asks to preprocess a set $\mathscr{O}$ of geometric objects in $\mathbb{R}^d$ into a data structure, so that one can quickly report or count all objects of $\mathscr{O}$ intersected by a query object $\gamma$, or just test whether $\gamma$ intersects any object of $\mathscr{O}$ at all. Intersection queries are generalization of range queries (in which the input objects are points) and point-enclosure queries (in which the query objects are points).

Intersection-searching problems in 2D have been studied since the early 1990s, see, e.g., [7] and surveys [1, 3], but these problems mostly reduce to 2D or 3D range searching. In general, intersection-searching queries can be formulated as semi-algebraic range queries or point-enclosure queries in an appropriate parametric space, but the storage and query time are large because the parametric space tends to be much higher dimensional than the ambient space [2]. For example, using semi-algebraic range searching data structures and multi-level partition trees based on geometric cuttings (see e.g. [13]), a line-intersection query, and its generalizations such as segment-intersection and ray-shooting queries, amid $n$ triangles or balls in $\mathbb{R}^3$ can be answered in $O^*(n^{3/4})$ time using $O^*(n)$ storage, in $O(\log n)$ time using $O^*(n^4)$ storage, or in $O^*(n/s^{1/4})$ time using $O^*(s)$ storage, for any $n \le s \le n^4$, by combining the first two solutions [13, 14].[1] Recently, Ezra and Sharir [8] proposed a new approach for answering ray-shooting queries amid triangles in $\mathbb{R}^3$, using the polynomial-partitioning scheme by Guth [9]. This approach was extended to 3D intersection-searching in a fairly general setting by Agarwal *et al.* [2].

Analogous to halfspace-emptiness and halfspace-reporting queries, intersection-detection and intersection-reporting queries in some cases can be answered more quickly than intersection-counting queries using the concept of shallow cutting [11]. For example, a line/segment intersection-detection query amid $n$ balls in $\mathbb{R}^3$ can be answered in $O^*(n/s^{1/3})$ time using $O^*(s)$ storage, for $n \le s \le n^3$ [12, 13, 14, 16], while the best known data structure for answering intersection-counting queries takes $O^*(n/s^{1/4})$ time, as mentioned above.

**Our results.** In this paper we make progress toward intersection queries between lines and unit balls in $\mathbb{R}^3$. Our first main result (Sections 2–4) is a linear-size data structure for answering line-intersection queries amid unit balls in $\mathbb{R}^3$:

► **Theorem 1.** *Let $\mathscr{B}$ be a set of $n$ unit balls in $\mathbb{R}^3$. $\mathscr{B}$ can be preprocessed, in $O(n \log n)$ time, into a linear-size data structure so that for a query line $\ell$ in $\mathbb{R}^3$, a line-intersection-detection query can be answered in $O^*(n^{1/2})$ time, and a line-intersection-reporting query can be answered in additional $O(k)$ time, where $k$ is the output size.*

We preprocess the centers of $\mathscr{B}$ into a data structure for answering unit-cylinder range emptiness/reporting queries (Section 2). Our main observation is that if the centers lie in a narrow slab, the region bounded by two parallel planes within distance 2 from each other, then a query unit cylinder $C$ can be replaced by $O(1)$ *cylindrical prisms*, each of which is of the form $\tau \oplus \mathsf{r_u}$, where $\tau$ is a carefully chosen portion of $\partial C$, $\mathsf{u}$ is one of $O(1)$ canonically chosen directions in $\mathbb{R}^3$, and $\mathsf{r_u}$ is the ray emanating from the origin in the direction $\mathsf{u}$ (Section 3). An advantage of working with such cylindrical prisms is that we can combine the theory of lower envelopes of bivariate functions [15] with Matoušek's [11] shallow-cutting technique to construct a linear-size data structure with $O^*(\sqrt{n})$ query time. One stumbling block in applying his technique to our setting is the construction of the so-called *test set.* Roughly speaking, a test set is a small representative set of all query cylindrical prisms, in the sense that if the data structure has small query time for the test set, it also has a similar query time for any cylindrical prism. The construction of a test set $Q$ in [11] for half-space range searching heavily relies on the linearity of hyperplanes. Agarwal and Matoušek had proposed an approach for constructing a test set for semi-algebraic ranges [4], but unfortunately it gives a weaker bound on the query time. Sharir and Shaul [16] were able to overcome this challenge by proposing a different approach for constructing a test set, which is fairly general. We adapt their approach to our setting for constructing a desired test set (Section 4).

[1] As in the abstract, the $O^*(\cdot)$ notation hides subpolynomial factors, e.g., of the form $O(n^{\varepsilon})$, for arbitrarily small $\varepsilon > 0$, and their coefficients which depend on $\varepsilon$.

Our second main result (Section 5) is an $O^*(n^2)$-size data structure for answering fixed-radius neighbor queries amid a set of lines in $\mathbb{R}^3$:

▶ **Theorem 2.** *Let $\mathscr{L}$ be a set of $n$ lines in $\mathbb{R}^3$. $\mathscr{L}$ can be preprocessed, in time $O^*(n^2)$, into a data structure of size $O^*(n^2)$ that can answer in $O(\log n)$ time whether a query point in $\mathbb{R}^3$ lies within unit distance from any of the lines of $\mathscr{L}$. Reporting the subset of these lines costs additional $O(k)$ time, where $k$ is the output size.*

This problem is equivalent to answering point-enclosure queries amid a set of unit cylinders in $\mathbb{R}^3$. Using a two-dimensional geometric cutting, we reduce the problem to the case when the query point lies inside a narrow slab. We then replace each input cylinder with $O(1)$ cylindrical prisms and check whether the query point lies in any of the cylindrical prisms.

## 2 Unit-Cylinder Range Searching

Let $P \subset \mathbb{R}^3$ be a set of $n$ points in $\mathbb{R}^3$. We wish to preprocess $P$ into a linear-size data structure so that a range-emptiness or a range-reporting query for a unit cylinder $C$ can be answered quickly. For simplicity, we assume that the axis of $C$ is not parallel to the $yz$-plane. A similar but simpler data structure can answer queries for unit cylinders whose axes are parallel to the $yz$-plane; we omit the details from this version. Let $\mathbb{C}$ be the family of unit cylinders whose axes are not parallel to the $yz$-plane. A cylinder $C_p \in \mathbb{C}$ can be represented by a point $p = (p_1, p_2, p_3, p_4) \in \mathbb{R}^4$ where $(p_1, p_2)$ and $(p_3, p_4)$ are intersection points of the axis of $C_p$ with the planes $x = 0$ and $x = 1$, respectively. We thus identify $\mathbb{C}$ with $\mathbb{R}^4$.

We construct a two-level partition tree $\Psi$ on $P$, as follows. For a point $p \in P$, let $p^*$ be its $xy$-projection, and let $P^* = \{p^* \mid p \in P\}$. Without loss of generality, we assume that no two points in $P$ project to the same point. Our top-level tree is a two-dimensional partition tree, based on simplicial partition, and some of its nodes store a second-level partition tree.

Let $S$ be a set of $n$ points in $\mathbb{R}^2$, and let $r > 0$ be a parameter. A *simplicial* $(1/r)$*-partition* for $P$ with respect to the parameter $r$ is a collection $\Pi = \{(S_1, \Delta_1), ..., (S_m, \Delta_m)\}$, where $m \le r$ is an integer, such that (i) $\{S_1, ..., S_m\}$ is a partition of $S$ (into pairwise-disjoint subsets) satisfying $n/r \le |S_i| \le 2n/r$, for each $i$, and (ii) each $\Delta_i$ is a (possibly degenerate) triangle, referred to as a *cell*, that contains $S_i$. In general, the cells $\Delta_i$ need not be disjoint. The *crossing number* of $\Pi$ for a line $\ell$ in $\mathbb{R}^2$ is the number of its cells that are crossed by $\ell$. The crossing number of $\Pi$ is defined as the maximum crossing numbers over all lines $\ell$. Matoušek [11] described an algorithm for constructing a simplicial partition whose crossing number is $O(\sqrt{r})$. If $r = O(1)$, the running time of his algorithm is $O(n)$; see also [5].

We choose $r$ to be a sufficiently large constant. By constructing simplicial partitions recursively and stopping the recursion as soon as the number of points becomes smaller than some sufficiently large constant $n_0$, we construct a two-dimensional partition tree on $P^*$, which is the primary tree of $\Psi$. See [1, 5, 11] for details. Each node $v \in \Psi$ is associated with a cell $\Delta_v$ and a subset $P_v^* \subseteq P^* \cap \Delta_v$. If $v$ is the root then $\Delta_v = \mathbb{R}^2$ and $P_v^* = P^*$. Let $P_v = \{p \in P \mid p^* \in P_v^*\}$ be the subset of $P$ corresponding to $P_v^*$. Set $n_v = |P_v|$. Let $\Delta_v^{\uparrow} = \Delta_v \times \mathbb{R}$ be the vertical prism erected over the cell $\Delta_v$. Then $P_v \subset \Delta_v^{\uparrow}$.

The *width* of a planar point set $X$ is the minimum distance between two parallel supporting lines of $X$. We call a cell of $\Psi$ *narrow* if its width is at most 2 and *wide* otherwise. For a node $v \in \Psi$, if $\Delta_v$ is narrow, we build a second-level partition tree $\Sigma_v$ on $P_v$ for answering range queries with a unit cylinder, using the algorithm described in Section 3. By Theorem 12, $P_v$ can be preprocessed, in $O(n \log n)$ time, into a linear-size data structure so that an emptiness query for a unit cylinder can be answered in $O^*(n^{1/2})$ time. This data structure can also report all $k$ points of $P_v$ lying in a query cylinder in an additional $O(k)$ time. This completes the description of the data structure.

**Query procedure.** Let $C$ be a query unit cylinder whose axis is not parallel to the $z$-axis, and let $C^*$ be its $xy$-projection, which is a strip of width 2 bounded by two parallel lines. For simplicity, we describe the procedure answering the emptiness query with $C$. We visit $\Psi$ recursively in a top-down manner, starting from its root. Suppose we are at a node $v \in \Psi$. If $C^* \cap \Delta_v = \emptyset$, then we simply return. If $v$ is a leaf, then we check all points of $P_v$ and return yes if any of them lies in $C$ and no otherwise. So assume that $v$ is an internal node and $\Delta_v \cap C^* \neq \emptyset$. If $\Delta_v$ is narrow then we use the secondary data structure $\Sigma_v$ stored at $v$ to test whether $P_v \cap C \neq \emptyset$; see Section 3. On the other hand, if $\Delta_v$ is wide (i.e., its width is more than 2), then we recursively visit the children of $v$. Note that if $\Delta_v$ is wide then it intersects at least one of the two boundary lines of $C^*$. For the emptiness query, the query procedure can terminate as soon as a point of $P$ inside $C$ is found. But for the reporting query, we continue with recursive calls until we have reported all the points.

**Analysis.** The height of $\Psi$ is $O(\log n)$, and some of its nodes store a linear-size secondary structure and others use $O(1)$ space, so the overall size of $\Psi$ is $O(n \log n)$. A similar argument shows that the preprocessing time is $O(n \log^2 n)$.

Concerning the query time, we present the analysis for emptiness queries. Reporting queries can be analyzed in a similar manner, where we gain an additional factor of $O(k)$ in the query time. Denote by $Q(n)$ the maximum emptiness query time for the two-level data structure on a set of $n$ points. For $n \leq n_0$, $Q(n) = O(n)$. If $\Delta_v$ is a narrow cell then we use the secondary data structure stored at $v$ and answer an emptiness query in $O^*(\sqrt{n_v})$ time. On the other hand, if $\Delta_v$ is wide then as mentioned above, one of the boundary lines of $C^*$ intersects $\Delta_v$. Since the crossing number of a simplicial partition is $O(\sqrt{r})$, the query is answered recursively at $O(\sqrt{r})$ children. The query procedure spends $O^*(\sqrt{n_v})$ time for each of the remaining children of $v$. We therefore, obtain the following recurrence for $n > n_0$.

$$Q(n) = O(\sqrt{r})Q(2n/r) + O^*(rn^{1/2}),$$

Since $r$ is a sufficiently large constant, the solution is $Q(n) = O^*(n^{1/2})$.

The $\log n$ factor can be removed from the space and preprocessing time using a standard technique of storing the second-level structure at every $\varepsilon \cdot \log n$ level of the primary structure, for a sufficiently small constant $\varepsilon > 0$. This adds a factor of $O(n^\varepsilon)$ to the query time, which is subsumed by our $O^*(\cdot)$ notation. This completes the proof of Theorem 1.

## 3 Range Queries for Narrow Cells

Let $P \subset \mathbb{R}^3$ be a set of $n$ points lying in a vertical slab $\sigma$ of width at most 2, and let $\mathbb{C}$ be the set of all unit cylinders whose axes are not parallel to the $yz$-plane. In this section, we describe a data structure for answering range emptiness and reporting queries on $P$ with a unit cylinder in $\mathbb{C}$. Let $H^-, H^+$ be the two parallel boundary planes of $\sigma$ with $x(H^-) < x(H^+)$. Without loss of generality, we assume that $H^-, H^+$ are normal to the $x$-axis. We also assume that the width of $\sigma$ is at most $w_0 = \sin^2(1/16)$ because otherwise we partition $\sigma$ into $O(1)$ slabs, each of width at most $w_0$, and build a separate data structure for each of them.

We first show that a range query on such a set $P$ with a unit cylinder in $\mathbb{C}$ can be reduced to answering range queries with $O(1)$ "cylindrical prisms," each of which is erected in one of the $O(1)$ carefully chosen canonical directions; see Section 3.1 for a precise problem formulation and the reduction. We then apply the machinery developed in [11, 16] to build the desired data structure. As mentioned in Section 1, a critical ingredient of this machinery is the construction of a test set, which, roughly speaking, is a small-size representative set of query cylindrical prisms. As in [4, 16], each range in the test set is not a cylindrical prism but a *generalized cylindrical prism*, the union of an infinite family of cylindrical prisms. We describe the notion of test set in Section 3.2 but postpone its construction to Section 4. We finally adapt the machinery of [11] for answering range queries with cylindrical prisms.

## 3.1 Reduction to cylindrical-prism queries

Let $\mathbb{S}^2$ be the unit sphere of directions in $\mathbb{R}^3$. For a direction $\mathsf{u} \in \mathbb{S}^2$, let $\mathsf{r}_\mathsf{u}$ be the ray emanating from the origin in direction $\mathsf{u}$. Set $\bar{\mathsf{u}} = -\mathsf{u}$. For a point $p \in \mathbb{R}^3$, $p + \mathsf{r}_\mathsf{u}$ is the ray in direction $\mathsf{u}$ emanating from $p$, and $p - \mathsf{r}_\mathsf{u} = p + \mathsf{r}_{\bar{\mathsf{u}}}$ is the ray emanating from $p$ in direction $\bar{\mathsf{u}}$. Let $\kappa > 16\pi$ be a sufficiently large constant. By choosing two orthogonal families of $O(\kappa)$ great circles, we partition $\mathbb{S}^2$ into "spherical grid" cells so that the (spherical) distance between any two points within a grid cell is at most $\frac{\pi}{\kappa}$. Let $\mathscr{G}$ be the set of these $O(\kappa)$ great circles, and let $\mathscr{A}(\mathscr{G})$ be the grid formed by the arrangement of $\mathscr{G}$.

**Cylindrical patches and prisms.** Let $C \in \mathbb{C}$ be a unit cylinder with axis $\ell$. Let the unit circle $C^*$ be the orthogonal projection $\partial C$ on a two-dimensional plane orthogonal to $\ell$, i.e., $C = C^* \times \ell$. For a point $p^* \in C^*$, the line $\{p\} \times \ell \subset \partial C$ is called a *generator line* of $C$. A *cylindrical patch* $\tau \subset \partial C$ is a portion of $\partial C$ bounded by two of its generator lines, i.e., $\tau = \delta \times \ell$, where $\delta \subset C^*$ is a unit arc spanning less than a semi-circle. We partition $\partial C$ into a family $\mathscr{P}(C)$ of $O(1)$ *canonical patches* using the grid $\mathscr{A}(\mathscr{G})$, as follows. Each patch in $\mathscr{P}(C)$ is the maximal portion of $\partial C$ whose (inner) normals lie within the same grid cell of $\mathscr{A}(\mathscr{G})$. The normals of $C$ form a great circle $C^\perp$ orthogonal to $\ell$. The generator lines on $\partial C$ at which normals of $C$ are the intersection points of $C^\perp$ with the great circles of $\mathscr{G}$ form the boundary lines of the canonical patches. The portion of $\partial C$ between two consecutive boundary lines forms a canonical patch. By construction, the normals within a canonical patch vary by at most $\pi/\kappa$. Although $\mathscr{A}(\mathscr{G})$ has $O(\kappa^2)$ cells, a cylinder in $\mathbb{C}$ has only $O(\kappa)$ canonical patches.

**Good directions.** A direction $\mathsf{u} \in \mathbb{S}^2$ is called *good* for a canonical patch $\tau \in \mathscr{P}(C)$ if the following two conditions hold:

**(G1)** The angle between $\mathsf{u}$ and the (inner) normal of either of the planes $H^-, H^+$ does not lie in the range $[\frac{\pi}{2} - \frac{\pi}{\kappa}, \frac{\pi}{2} + \frac{\pi}{\kappa}]$, i.e., if $\mathsf{u} = (\mathsf{u}_x, \mathsf{u}_y, \mathsf{u}_z)$ then $|\mathsf{u}_x| \geq \sin\frac{\pi}{\kappa}$. Recall that the inner normals of $H^-, H^+$ are $(\pm 1, 0, 0)$. This condition says that $\mathsf{u}$ is not "nearly parallel" to the plane $H^-$ (or $H^+$).

**(G2)** The angle between $\mathsf{u}$ and the *inner* normal $\mathbf{n}_p$ for any $p \in \tau$ is at most $\frac{\pi}{2} - \frac{\pi}{\kappa}$, i.e., $\langle \mathbf{n}_p, \mathsf{u} \rangle \geq \sin\frac{\pi}{\kappa}$. This condition says that for any point $p \in \tau$, the ray $p + \mathsf{r}_\mathsf{u}$ enters $C$ and it is not "nearly parallel" to the tangent plane of $C$ at $p$.

(G1)–(G2) imply that for a point $p$ on a canonical patch $\tau$, $p + \mathsf{r}_\mathsf{u}$ enters $C$ and exits slab $\sigma$ before it exits $C$. We make this notion more precise in Lemma 5 below.

▶ **Lemma 3.** *There exists a constant $\delta := \delta(\kappa) \geq \frac{\pi}{\kappa}$ such that the set of good directions for any canonical patch $\tau(C)$ of a cylinder $C \in \mathbb{C}$ contains a spherical cap of radius $\delta$.*

**Proof.** We show that there is a direction $\mathsf{u}$ that is at least $\delta$ far away from all bad directions for $\tau$, which would imply the lemma.

Let $B^\oplus \subset \mathbb{S}^2$ be the set of all directions that are within (spherical) distance $\delta$ from a bad direction for $\tau$. $B^\oplus$ is the union of two sets $B_0^\oplus, B_1^\oplus$, formed by the complement of the two aforementioned conditions (G1) and (G2) of good directions. More precisely,

**(I)** Let $B_0$ be the set of all points on $\mathbb{S}^2$ that lie within (spherical) distance $\frac{\pi}{\kappa}$ from the great circle normal to the vector $(1,0,0)$, i.e., parallel to the planes $H^-, H^+$. $B_0^{\oplus}$ is the set of points that lie within distance $\delta$ from $B_0$. The area of $B_0^{\oplus}$ is at most $4\pi\sin(\frac{\pi}{\kappa}+\delta)$.

**(II)** Let $B_1$ be the set of directions that make an angle of more then $\frac{\pi}{2}-\frac{\pi}{\kappa}$ from some point of $\tau$. Since the normals within $\tau$ vary by at most $\pi/\kappa$, $B_1$ is a spherical cap of angular opening at most $\frac{\pi}{2}+\frac{2\pi}{\kappa}$. $B_1^{\oplus}$ is the set of points on $\mathbb{S}^2$ within distance $\delta$ from $B_1$ and thus spherical cap of radius $\frac{\pi}{2}+\frac{2\pi}{\kappa}+\delta$. Hence, the area of $B_1^{\oplus}$ is at most $2\pi+2\pi\sin(\frac{2\pi}{\kappa}+\delta)$.

Summing these areas, we obtain

$$\mathrm{Area}(B^{\oplus}) \le 2\pi\left(1+2\sin\left(\frac{\pi}{\kappa}+\delta\right)+\sin\left(\frac{2\pi}{\kappa}+\delta\right)\right) < 4\pi,$$

provided we choose $\delta=\frac{\pi}{\kappa}$ and $\kappa \ge 16\pi$. As such $\mathbb{S}^2 \setminus B^{\oplus} \neq \emptyset$, and there exists a direction $\mathsf{u}$ such that all directions in the spherical cap of radius $\delta$ centered at $\mathsf{u}$ are good for $\tau$. ◀

In the following, we set $\delta=\frac{\pi}{\kappa}$. Let $B_0 \subset \mathbb{S}^2$ be, as above, the set of directions that violate the condition (G1). We choose $\mathscr{Z} \subset \mathbb{S}^2 \setminus B_0$ to be a set of $O(1/\delta^2)$ points that is a $\delta$-*net* for $\mathbb{S}^2 \setminus B_0$, i.e., for any point on $\mathbb{S}^2 \setminus B_0$, there is a point in $\mathscr{Z}$ within distance $\delta$. For simplicity, we assume that $\mathscr{Z}$ is centrally symmetric. Lemma 3, the definition of $B_0$ and the choice of $\delta$ immediately imply the following:

▶ **Corollary 4.** *For any point $\mathsf{v} \in \mathbb{S}^2$, there is a point $\mathsf{u} \in \mathscr{Z}$ within distance $\frac{2\pi}{\kappa}$ from $\mathsf{v}$.*

**Reduction to cylindrical prisms.** We first prove a key property of good directions:

▶ **Lemma 5.** *Let $C \in \mathbb{C}$ be a unit cylinder, let $\tau \in \mathscr{P}(C)$ be a canonical patch, and let $\mathsf{u}$ be a good direction for $\tau$. Then any line parallel to $\mathsf{u}$ intersects $\tau$ in at most one point. Moreover, for any point $p \in \tau \cap \sigma$, the ray $p + \mathsf{r}_{\mathsf{u}}$ exits $\sigma$ before exiting $C$.*

**Proof.** If a line $\ell$ parallel to $\mathsf{u}$ intersects $\tau$ twice, then by the Intermediate Value Theorem, $\tau$ must have a tangent in the $\mathsf{u}$-direction, which contradicts property (G2) of $\mathsf{u}$ being a good direction. Hence, $\ell$ intersects $\tau$ at most once.

We next prove the second assertion of the lemma. Let $p'$ be the other intersection point of $\partial C$ and the ray $p + \mathsf{r}_{\mathsf{u}}$. We observe that $|pp'|$ is minimized when $pp'$ is orthogonal to the axis of $C$ and forms an angle of $\frac{\pi}{\kappa}$ with the tangent plane to $C$ at $p$. It thus follows that $|pp'| \ge 2\sin\left(\frac{\pi}{\kappa}\right)$. On the other hand, by property (G1) of good direction, $\mathsf{r}_{\mathsf{u}}$ forms an angle of at most $\frac{\pi}{2}-\frac{\pi}{\kappa}$ with the outer normal $\mathbf{n}$ to the plane $H^-$ (or $H^+$). Consequently, the length of the projection of $pp'$ on $\mathbf{n}$ is at least

$$|pp'|\cos\left(\frac{\pi}{2}-\frac{\pi}{\kappa}\right) = |pp'|\sin\left(\frac{\pi}{\kappa}\right) \ge 2\sin^2\left(\frac{\pi}{\kappa}\right) \ge 2\sin^2\left(\frac{1}{16}\right) > w_0.$$

Thus $p'$ lies outside $\sigma$, which completes the proof of the lemma. ◀

For a direction $\mathsf{u} \in \mathscr{Z}$, let $\mathscr{P}_{\mathsf{u}}(C) \subset \mathscr{P}(C)$ denote the subset of canonical patches of $C$ for which $\mathsf{u}$ is a good direction. We construct a *canonical prism* $\tau_{\mathsf{u}}^{\uparrow}$ on every patch $\tau \in \mathscr{P}_{\mathsf{u}}(C)$. We construct $O(1)$ canonical prisms for every patch $\tau \in \mathscr{P}(C)$, one for every direction $z$ that is good for $\tau$. The construction of the prism is somewhat delicate because we wish to meet two conflicting constraints: (i) we want to ensure that $\tau_{\mathsf{u}}^{\uparrow} \cap \sigma$ lies inside $C \cap \sigma$, and (ii) the union of the canonical prisms over all canonical patches and over all their good directions in

$\mathscr{Z}$ covers $C \cap \sigma$. At a high level, we carefully clip $\tau$ by a constant-complexity semi-algebraic curve lying on $\tau$ so that any generator line of $\tau$ intersects the curve exactly once. Let $\hat{\tau}_u \subset \tau$ be the *clipped patch* of $\tau$ with respect to direction $\mathsf{u}$. We set $\tau_{\mathsf{u}}^{\uparrow} := \hat{\tau} \oplus \mathsf{r}_{\mathsf{u}}$. We now describe the construction of $\hat{\tau}$.

We fix a direction $\mathsf{u} \in \mathscr{Z}$. For a point $p \in \tau$, let $e_{\mathsf{u}}(p) = (p + \mathsf{r}_{\mathsf{u}}) \cap \sigma$ be the segment of the ray $p + \mathsf{r}_{\mathsf{u}}$ that lies inside $\sigma$; $e_{\mathsf{u}}(p)$ may be empty. Let $\phi_{\mathsf{u}}(p)$ be the other endpoint of $e_{\mathsf{u}}(p)$ if $e_{\mathsf{u}}(p) \neq \emptyset$. We set

$$\hat{\tau}_{\mathsf{u}} := \{p \in \tau \mid e_{\mathsf{u}}(p) \subset C\}, \quad \tau_{\mathsf{u}}^{\uparrow} = \hat{\tau}_{\mathsf{u}} \oplus \mathsf{r}_{\mathsf{u}}, \quad \text{and} \quad \partial\hat{\tau}_{\mathsf{u}} = \{p \in \tau \mid \phi(p) \in \partial C\}. \tag{1}$$

(Here we use the convention that if $e_{\mathsf{u}}(p) = \emptyset$ then it lies inside $C$.) The patch $\tau$ is clipped along the arc $\partial\tau_{\mathsf{u}}$. We note that the points $\phi_{\mathsf{u}}(p)$ for $p \in \partial\tau_{\mathsf{u}}$ lie on $\partial C \cap H^-$ (resp. $\partial C \cap H^+$) if the $x$-component of $\mathsf{u}$ is positive (resp. negative); see Figure 1.

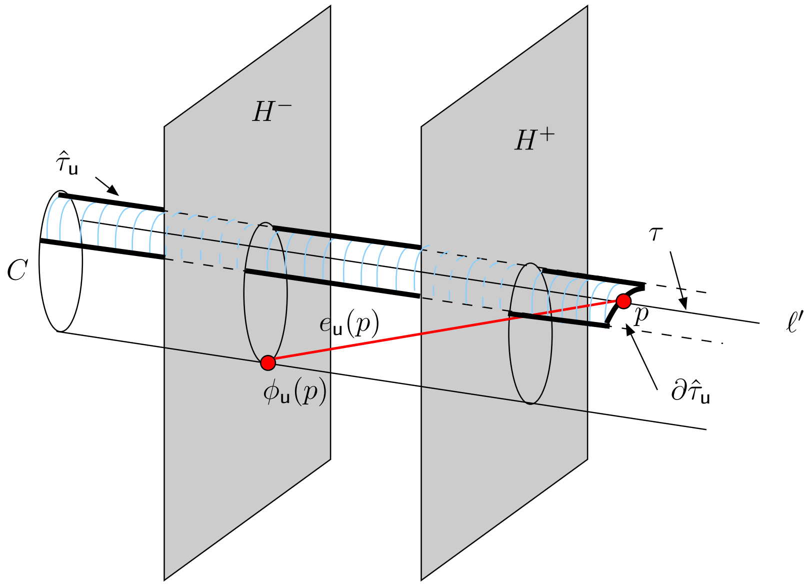


**Figure 1** A clipped canonical patch $\hat{\tau}_{\mathsf{u}}$ and its boundary arc $\partial\tau_{\mathsf{u}}$; segment $e_{\mathsf{u}}(p)$ and its endpoint $\phi_{\mathsf{u}}(p)$ for a point $p \in \tau$.

Fix one of the generator lines $\ell'$ of $\tau$. For any point $p \in \ell'$, the intersection point $\bar{p} = p + \mathsf{r}_{\mathsf{u}}$ with $\partial C$ lies on another generator line of $C$, say, $\ell''$. As we translate $p$ along $\ell'$, $\bar{p}$ also translates along $\ell''$ (with the segment $p\bar{p}$ being in direction $\mathsf{u}$), and thus there is a unique point $p_{\ell'} \in \ell'$ for which $\bar{p}_{\ell'} = \phi_{\mathsf{u}}(p_{\ell'})$, i.e., $p_{\ell'} \in \partial\hat{\tau}_{\mathsf{u}}$. The following lemma easily follows from the convexity of $C$:

▶ **Lemma 6.**

**(i)** *Each generator line $\ell'$ of $\tau$ contains exactly one point $p_{\ell'}$ of $\partial\hat{\tau}_{\mathsf{u}}$. If the $x$-component of $\mathsf{u}$ is positive (resp. negative), then $\{p \in \ell \mid x(p) \geq x(p_{\ell'})\}$ (resp. $\{p \in \ell \mid x(p) \leq x(p_{\ell'})\}$ is the portion of $\ell'$ that lies in $\hat{\tau}_{\mathsf{u}}$.*

**(ii)** *$\tau_{\mathsf{u}}^{\uparrow} \cap \sigma \subset C \cap \sigma$.*

We construct the canonical prisms for every patch in $\mathscr{P}_{\mathsf{u}}(C)$, and we repeat this step for all directions in $\mathscr{Z}$. Finally, we set

$$\Lambda(C) := \bigcup_{\mathsf{u} \in \mathscr{Z}} \{\tau_{\mathsf{u}}^{\uparrow} \mid \tau \in \mathscr{P}_{\mathsf{u}}(C)\}$$

to be the set of all canonical prisms erected over the patches of $\mathscr{P}_{\mathsf{u}}(C)$, and set

$$\mathscr{U}(C) := \bigcup_{\tau^{\uparrow} \in \Lambda(C)} \tau^{\uparrow}, \tag{2}$$

to be the union of these canonical prisms.

▶ **Lemma 7.** *For any cylinder $C \in \mathbb{C}$, $\mathscr{U}(C) \cap \sigma = C \cap \sigma$.*

**Proof.** It follows immediately from Lemma 6 that $\mathscr{U}(C) \cap \sigma \subseteq C \cap \sigma$. Therefore it remains to prove that $\mathscr{U}(C) \cap \sigma \supseteq C \cap \sigma$. Let $q$ be a point in $C \cap \sigma$. We show that there exists a direction $\mathsf{u} \in \mathscr{Z}$ and a patch $\tau \in \mathscr{P}_{\mathsf{u}}(C)$ such that $q \in \tau_{\mathsf{u}}^{\uparrow}$.

Let $q^*$ be the projection of $q$ on the axis of $C$. Let $\mathsf{u} \in \mathscr{Z}$ be the direction closest to $q^*q$, and let $\bar{\mathsf{u}} = -\mathsf{u}$ which is also in $\mathscr{Z}$. By the construction of $\mathscr{Z}$, $\mathsf{u}, \bar{\mathsf{u}}$ satisfy (G1). Let $q'$ (resp. $q''$) be the intersection point of the ray $q + \mathsf{r}_{\mathsf{u}}$ (resp. $q + \mathsf{r}_{\bar{\mathsf{u}}} = q - \mathsf{r}_{\mathsf{u}}$) with $\partial C$, and let $\tau'$ (resp. $\tau''$) be the canonical (unclipped) patch of $C$ containing $q'$ (resp. $q''$). Let $\ell' \subset \tau'$ (resp. $\ell'' \subset \tau''$) be the generator line of $C$ containing $q'$ (resp. $q''$). See Figure 2.

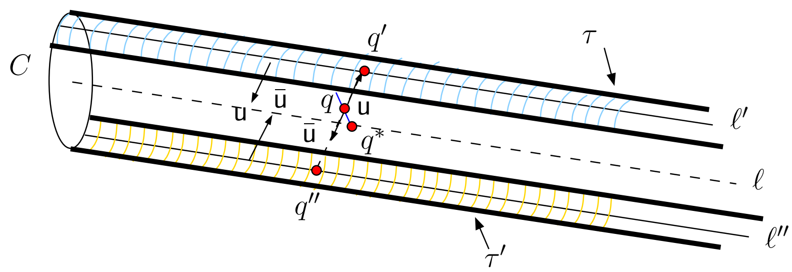


**Figure 2** Points $q, q^*, q'$, and $q''$; generator lines and patches containing $q'$ and $q''$.

We first claim that $\bar{\mathsf{u}}$ is a good direction for $\tau'$. By Corollary 4, $\angle(qq^*, \bar{\mathsf{u}}) \leq \frac{2\pi}{\kappa}$. Furthermore, $\angle(qq^*, \mathbf{n}_{q'}) \leq \angle(qq^*, \bar{\mathsf{u}}) \leq \frac{2\pi}{\kappa}$. Therefore

$$\angle(\mathbf{n}_{q'}, \bar{\mathsf{u}}) \leq \angle(qq^*, \bar{\mathsf{u}}) + \angle(qq^*, \mathbf{n}_{q'}) \leq \tfrac{4\pi}{\kappa}.$$

However, a direction that violates (G2) for patch $\tau$ (i.e., it is in $B_1$) makes an angle of at least $\frac{\pi}{2} - \frac{2\pi}{\kappa}$ with $\mathbf{n}_{q'}$, which is more than $\frac{4\pi}{\kappa}$ by our choice of $\kappa$. Hence, $\bar{u}$ is a good direction for $\tau'$. A similar argument shows that $\mathsf{u}$ is a good direction for $\tau''$. If $q' \in \hat{\tau}'$ then $q$ lies in the canonical prism $\tau'^{\uparrow}_{\bar{\mathsf{u}}}$, and similarly if $q'' \in \hat{\tau}''$ then $q$ lies in the prism $\tau''^{\uparrow}_{\mathsf{u}}$. We therefore argue that at least one of these conditions holds.

We claim that if $q'$ does not lie in the clipped patch $\hat{\tau}'$, then $q''$ lies in the clipped patch $\hat{\tau}''$. Without loss of generality, assume that the $x$-component of $\mathsf{u}$ is positive. Let $G$ be the plane spanned by $\ell'$ and $\ell''$; the segment $q'q''$ lies in $G$. Let $g^- = G \cap H^-$ and $g^+ = G \cap H^+$ be the intersection lines of $G$ with the boundary planes of $\sigma$; $x(g^+) > x(g^-)$. Then $G \cap \sigma$ is the strip lying between the parallel lines $g^-, g^+$. By definition, $q \in G \cap \sigma$. Let $w' = \ell' \cap \partial\hat{\tau}'_{\bar{\mathsf{u}}}$ (resp. $w'' = \ell'' \cap \partial\hat{\tau}''_{\mathsf{u}}$) be the point on $\ell'$ (resp. $\ell'$) that lies on the boundary arc $\partial\hat{\tau}'_{\bar{\mathsf{u}}}$ (resp. $\partial\hat{\tau}''_{\mathsf{u}}$). See Figure 3.

Since the $x$-component of $\mathsf{u}$ is assumed to be positive (and thus the $x$-component of $\bar{\mathsf{u}}$ is negative), by Lemma 6,

$$\hat{\tau}'_{\bar{\mathsf{u}}} \cap \ell' = \{p \in \ell' \mid x(p) \leq x(w')\} \quad \text{and} \quad \hat{\tau}''_{\mathsf{u}} \cap \ell'' = \{p \in \ell'' \mid x(p) \geq x(w'')\}. \tag{3}$$

Let $\bar{w}' = \phi_{\bar{\mathsf{u}}}(w')$ and $\bar{w}'' = \phi_{\mathsf{u}}(w'')$ be the other endpoints of the segments $e_{\bar{\mathsf{u}}}(w')$ and $e_{\mathsf{u}}(w'')$, respectively. By definition, $\bar{w}' = \ell'' \cap g^-$ and $\bar{w}'' = \ell' \cap g^+$. Furthermore, the segments

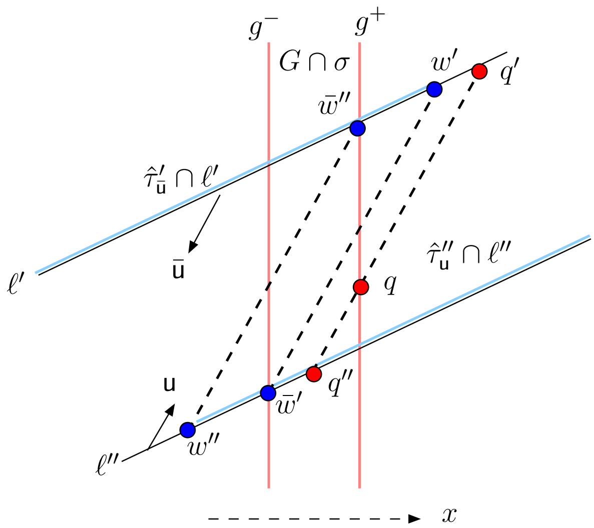


**Figure 3** Illustration of at least one of $q'$ and $q''$ lying on the clipped patch.

$q'q''$, $w'\bar{w}'$, and $w''\bar{w}''$ are parallel to each other, with their endpoints lying on $\ell'$ and $\ell''$. By Lemma 5, $x(w') > x(g^+) = x(\bar{w}'')$ because the ray $\bar{w}' + \mathsf{r}_{\mathsf{u}} \subset G$ exits $\sigma$ before exiting $C$, and similarly $x(w'') < x(g^-) = x(\bar{w}')$. See Figure 3. If $q' \notin \hat{\tau}'_u$ then by (3), $x(q') > x(w')$ and thus $x(q') > x(\bar{w}'')$. Since $q'q''$ and $w''\bar{w}''$ are parallel segments, we conclude that $x(q'') > x(w'')$ and therefore by (3), $q'' \in \ell'' \cap \hat{\tau}''_{\mathsf{u}}$. Hence, if $q' \notin \hat{\tau}'_{\mathsf{u}}$ then $q'' \in \hat{\tau}''_{\mathsf{u}}$. This completes the proof of the lemma. ◀

Recall that $P \subset \sigma$, therefore by Lemma 7, we can answer an emptiness (or reporting) query with a unit cylinder $C \in \mathbb{C}$ by answering emptiness (or reporting) queries with all cylindrical prisms in $\Lambda(C)$ (see (2)). Fix a grid cell $\varphi \in \mathscr{A}(\mathscr{G})$ and a direction $\mathsf{u} \in \mathscr{Z}$ that is good for patches corresponding to $\varphi$. Let $\mathbb{C}_{\mathsf{u},\varphi}$ be the set of all cylindrical prisms $\tau^{\uparrow}_{\mathsf{u}}$ erected in direction $\mathsf{u}$ over the canonical patches $\tau$ corresponding to the grid cell $\phi$ of unit cylinders in $\mathbb{C}$. We build a separate partition tree $\mathscr{T}_{\mathsf{u},\varphi}$ for answering range queries with prisms in $\mathbb{C}_{\mathsf{u},\varphi}$. In the rest of the section, we describe how we build $\mathscr{T}_{\mathsf{u},\varphi}$ by adapting the approach in [16].

## 3.2 Test set for cylindrical prisms

Throughout this section, let $r > 1$ be a fixed parameter, which we will choose to be a sufficiently large constant. We call a semi-algebraic set $\triangle$ $(1/r)$-*shallow* (or simply *shallow* if the value of $r$ is clear from the context) with respect to $P$ if $|P \cap \triangle| \le n/r$.

Following the terminology in [16], we call a family $\mathscr{Q}$ of constant-complexity semi-algebraic sets, which we will refer to as *generalized prisms*, a test set for $\mathbb{C}_{\mathsf{u},\phi}$ with respect to $P$ and $r$ if the following properties hold:

**(C1)** *Compactness*: $|\mathscr{Q}| = r^{O(1)}$.

**(C2)** *Shallowness*: Each generalized prism $\pi^{\uparrow} \in \mathscr{Q}$ is $(1/r)$-shallow with respect to $P$.

**(C3)** *Containment*: Each $(1/r)$-shallow cylindrical prism $\tau^{\uparrow}_{\mathsf{u}} \in \mathbb{C}_{\mathsf{u},\phi}$ is contained in a single generalized cylindrical prism $\pi^{\uparrow}$ of $\mathscr{Q}$, i.e., $\tau^{\uparrow}_{\mathsf{u}} \subseteq \pi^{\uparrow}$.

**(C4)** *Efficiency*: There exists a small bound on the associated function $\zeta(m)$, bounding the size of a partition of the *free space*, the complement of the union, of any subset of $m$ generalized cylindrical prisms of $\mathscr{Q}$ into elementary cells.

Each set in $\mathscr{Q}$ will be the union of cylindrical prisms erected in direction $\mathsf{u}$ over an infinite family of clipped canonical patches – see Section 4 for details. Informally, properties (C1)-(C3) imply that instead of considering the whole family of cylindrical prisms in $\mathbb{C}_{\mathsf{u},\phi}$, we can consider a small finite set $\mathscr{Q}$ of "representative queries" from a more general set, each of which is shallow with respect to $P$, such that if the partition tree we build has a small query time for a range in $\mathscr{Q}$ then it also has roughly the same query time for any cylindrical prism. Property (C4) bounds the query time for a range in the test set. We describe, in Section 4, the construction of a test set $\mathscr{Q}$ of size $O(r^4)$ with $\zeta(m) = O^*(m^2)$ (cf. Lemma 13), and $\mathscr{Q}$ can be constructed in $O(n)$ time if $r$ is a constant.

## 3.3 Data structure

With a small-size test set at hand, we are now ready to describe the algorithm for constructing an elementary-cell partition of $P$ and the partition tree by closely following the mechanism in [11, 16]. Let $P$ and $r$ be the same as above.

**Geometric cuttings.** Given a family $\Gamma$ of $n$ constant-complexity semi-algebraic sets in $\mathbb{R}^d$, a weight function $\omega : \Gamma \to \mathbb{R}^+$, and a parameter $r > 1$, a $(1/r)$-*cutting* for $\Gamma$ is a partition of space (or a portion thereof) into elementary cells, such that total weight of sets crossed by each cell is at most $\omega(\Gamma)/r$. The following lemma is taken from [16].

▶ **Lemma 8.** *Let $\Gamma$ be a collection of $n$ semi-algebraic sets of constant complexity in $\mathbb{R}^d$, let $\omega : \Gamma \to \mathbb{R}^+$ a weight function, and $r > 1$ a parameter. Assume that the free space of any subset of $m$ sets in $\Gamma$ can be partitioned into at most $\zeta(m)$ elementary cells, where $\zeta(\cdot)$ is a super-linear function. Then there exists a $(1/r)$-cutting $\Xi$ of $\Gamma$ of size $O(\zeta(r))$ that covers the free space of $\Gamma$. Furthermore, the free space of $\Xi$ is covered by the union of $O(r)$ sets of $\Gamma$. $\Xi$ can be constructed in $O(n)$ time if $r$ is a constant.*

By combining Lemmas 8 and 13, we obtain the following:

▶ **Corollary 9.** *Let $\mathscr{Q}$ be a collection of $n$ generalized prisms in $\mathbb{R}^3$ satisfying (C1)–(C4). Let $\omega : \mathscr{Q} \to \mathbb{R}^+$ be a weight function, and let $r \in [1, n]$ be a parameter. There exists a $(1/r)$-cutting $\Xi$ of $\mathscr{Q}$ of size $O^*(r^2)$ that covers the free space of $\mathscr{Q}$. Furthermore, the free space of $\Xi$ can be covered by $O(r)$ generalized prisms in $\mathscr{Q}$. $\Xi$ can be constructed in $O(n)$ time if $r$ is a constant.*

**Elementary-cell partition and partition tree.** Let $P$ be a set of $n$ points in $\mathbb{R}^3$, and let $r > 1$ be a parameter. We extend the notion of simplicial partition reviewed in Section 2 to answering queries with cylindrical prisms in $\mathbb{C}_{\mathsf{u},\phi}$, as follows.

An *elementary-cell* $(1/r)$-*partition* of $P$ is a collection $\Phi = \{(P_1, \triangle_1), ..., (P_m, \triangle_m)\}$, for some integer $m = O(r)$, such that (i) each $\triangle_i$ is an *elementary cell*, (ii) $\{P_1, ..., P_m\}$ is a partition of $P$, s.t. $P_i \subset \triangle_i$, and $n/r \leq |P_i| \leq 2n/r$. The cells $\triangle_i$ may overlap. The *crossing number* of $\Phi$ for a range $R$ is the number of elementary cells of $\Phi$ crossed by $R$, i.e., the number of elementary cells that intersect $\partial R$. The following lemma is a slight adaptation of the argument in [16] and its proof exploits Corollary 9:

▶ **Lemma 10.** *Let $P$ be a set of $n$ points in $\mathbb{R}^3$ lying in the slab $\sigma$ of width at most $\sin^2(1/16)$, let $r > 1$ be a fixed parameter, and let $\mathscr{Q}$ be a family of generalized prisms satisfying (C1)–(C4). Then there exists an elementary-cell $(1/r)$-partition $\Phi$ of $P$ such that the crossing number of $\Phi$ for any range in $\mathscr{Q}$ is $O(r/\zeta^{-1}(r) + \log r \log |\mathscr{Q}|)$. $\Phi$ can be computed in $O(n)$ time if $r$ is a constant.*

Plugging Lemma 13 and Corollary 9 in Lemma 10, we obtain the following corollary:

▶ **Corollary 11.** *let $P$ be a set of $n$ points lying in the slab $\sigma$ of width at most $\sin^2(1/16)$, and let $r \geq 1$ be a fixed parameter. Then there exists an elementary cell $(1/r)$-partition $\Phi$ of $P$ such that the crossing number of $\Phi$ for any $(1/r)$-shallow cylindrical prism in $\mathbb{C}_{\mathsf{u},\phi}$ is $O^*(r^{1/2})$. $\Phi$ can be constructed in $O(n)$ time if $r$ is a constant.*

By applying Corollary 11 recursively in a standard manner–see [11, 16]– we can build the partition tree $\mathscr{T}_{\mathsf{u},\varphi}$ of size $O(n)$ in $O(n \log n)$ time for answering emptiness or reporting queries with the cylindrical prisms in $\mathbb{C}_{\mathsf{u},\varphi}$. Since the crossing number of the elementary-cell partition is $O^*(r^{1/2})$, the query time for an emptiness query is $O^*(n^{1/2})$, and all $k$ points lying in a query range can be reported in an additional $O(k)$ time. Omitting all the details, which can be found in [11, 16], we obtain the following result:

▶ **Theorem 12.** *Let $P$ be a set of $n$ points in $\mathbb{R}^3$ lying inside a vertical slab of width at most 2. $P$ can be preprocessed, in $O(n \log n)$ time, into a data structure of size $O(n)$, so that for a (unit) cylinder $C \in \mathbb{C}$, an emptiness query can be answered in $O^*(n^{1/2})$ time, and all $k$ points of $C \cap P$ can be reported in additional $O(k)$ time.*

## 4 Test-Set Construction

We now describe the construction of a test set for cylindrical prisms in $\mathbb{C}_{\mathsf{u},\varphi}$, for a fixed grid cell $\varphi \in \mathscr{A}(\mathscr{G})$ and $\mathsf{u} \in \mathscr{Z}$, that satisfies (C1)–(C4). Recall that the space of cylinders in $\mathbb{C}$ is identified with $\mathbb{R}^4$. For a fixed $\varphi$ and $\mathsf{u}$, a cylindrical prism $\pi^\uparrow$ is uniquely defined by the cylinder $C \in \mathbb{C}$ whose boundary contains $\pi^\uparrow$, so the space of cylindrical prisms in $\mathbb{C}_{\mathsf{u},\varphi}$ can also be identified with $\mathbb{R}^4$. For a prism $p \in \mathbb{R}^4$, let $\pi_p^\uparrow$ be the cylindrical prism defined by $p$, i.e., the prism erected in direction $\mathsf{u}$ over the canonical patch of the unit cylinder $C_p \in \mathbb{C}$ corresponding to the grid cell $\varphi$. If $C_p$ does not contain any canonical patch corresponding to $\varphi$, then we regard $\pi_p^\uparrow$ as an empty set.

For a point $a \in \mathbb{R}^3$, we define the region $\mathsf{R}_a := \{p \in \mathbb{R}^4 \mid a \in \pi_p^\uparrow\}$ to be the locus of all points in $\mathbb{R}^4$ representing cylindrical prisms that contain $a$. $\mathsf{R}_a$ is a semi-algebraic set of constant complexity. We choose a random subset $N \subseteq P$ of $O(r \log r)$ points, with an appropriate constant of proportionality. We then form the set of regions $\mathscr{R} := \{\mathsf{R}_p \mid p \in N\}$ and construct their arrangement $\mathscr{A}(\mathscr{R})$. Set $k = c \ln r$, where $c > 0$ is an appropriate constant of proportionality. Let $\mathscr{A}_{\leq k}(\mathscr{R})$ be the set of all points of $\mathscr{A}(\mathscr{R})$ at *level* at most $k$, that is, these points represent all cylindrical prisms of $\mathbb{C}_{\mathsf{u},\varphi}$ that contain at most $k$ points of $P$. We compute the vertical decomposition of the cells of $\mathscr{A}_{\leq k}(\mathscr{R})$ [15], which decomposes each cell of $\mathscr{A}_{\leq k}(\mathscr{R})$ into elementary cells (each of which, in fact, is a pseudo-prism). Let $\mathscr{A}_{\leq k}^{\nabla}(\mathscr{R})$ be set of resulting elementary cells; $|\mathscr{A}_{\leq k}^{\nabla}(\mathscr{R})| = O^*(r^4)$ [10].

For an elementary cell $\triangle \in \mathscr{A}_{\leq k}^{\nabla}(\mathscr{R})$, let $\pi_\triangle^\uparrow = \bigcup_{p \in \triangle} \pi_p^\uparrow$ be the *generalized (cylindrical) prism*, which is the union of an infinite family of cylindrical prisms defined by the points in $\triangle$. The generalized prism $\pi^\uparrow$ is a constant-complexity semi-algebraic set that is unbounded in direction $\mathsf{u}$ and has the property that for any $q \in \pi^\uparrow$, $q + \mathsf{r}_\mathsf{u} \subseteq \pi^\uparrow$. We set $\Pi^\uparrow := \Pi^\uparrow(\mathscr{R}) = \{\pi_\triangle^\uparrow \mid \triangle \in \mathscr{A}_{\leq k}^{\nabla}(\mathscr{R})\}$ to be the family of $O^*(r^4)$ generalized prisms corresponding to the cells in $\mathscr{A}_{\leq k}^{\nabla}(\mathscr{R})$. Following a straightforward argument, as in [16], it can be shown that $\Pi^\uparrow$ satisfies (C1)–(C3). It thus suffices to prove (C4), namely, that the free space $\mathscr{K}(\mathscr{P}^\uparrow)$ of any subset $\mathscr{P}^\uparrow \subseteq \Pi^\uparrow$ of $m$ generalized prisms can be partitioned into $O^*(m^2)$ elementary cells.

In the following, without loss of generality, we assume that $\mathsf{u} = (0, 0, 1)$. Let $\mathscr{P}^\uparrow \subseteq \Pi$ be a subset of $m$ generalized prisms. Let $\pi$ denote the lower boundary of a $\pi^\uparrow \in \mathscr{P}^\uparrow$, i.e., the set of points $p \in \pi^\uparrow$ for which the open ray $p - \mathsf{r}_\mathsf{u}$ emanating from $p$ in the $(-u)$-direction is disjoint from $\pi^{\uparrow}$; $\pi^{\uparrow} = \pi \oplus \mathsf{r_u}$. We refer to $\pi$ as a *generalized (cylindrical) patch*, which is a constant-complexity two-dimensional $xy$-monotone semi-algebraic set. The patch $\pi$ can be viewed as the graph of a partially defined bivariate function, also denote by $\pi$. (The value of the function is set to $+\infty$ for every point $(x, y) \in \mathbb{R}^2$ at which $\pi$ is not defined.) Set $\mathscr{P} = \{\pi \mid \pi^{\uparrow} \in \mathscr{P}^{\uparrow}\}$. The *lower envelope* of $\mathscr{P}$ is defined as the graph of the function

$$\mathscr{E}(x, y) = \min_{\pi \in \mathscr{P}} \pi(x, y),$$

which, with a slight abuse of notation, is also denoted by $\mathscr{E}$. It induces a partition of $\mathbb{R}^2$ into maximal connected regions such that $\mathscr{E}$ is attained by a single generalized patch of $\mathscr{P}$ (or by none of them) over the interior of each such region. The boundary of such a region consists of points at which $\mathscr{E}$ is attained by at least two of the generalized patches in $\mathscr{P}$, or by the boundary of one of them. Let $\mathscr{M}$ denote this planar subdivision, called the *minimization diagram* of $\mathscr{P}$. The combinatorial complexity of $\mathscr{E}$ and $\mathscr{M}$ is the number of faces of all dimensions in $\mathscr{M}$, and it is bounded by $O^*(m^2)$ [15]. The free space $\mathscr{K}(\mathscr{P}^{\uparrow})$ is the set of points in $\mathbb{R}^3$ lying below the lower envelope $\mathscr{E}$.

We partition $\mathscr{K}(\mathscr{P}^{\uparrow})$ into elementary cells, as follows. We first compute the two-dimensional vertical decomposition of every face $f$ of $\mathscr{M}$, which partitions $f$ into pseudo-trapezoids. Let $\mathscr{M}^{\triangledown}$ denote the resulting refinement of $\mathscr{M}$. By construction, the same function of $\mathscr{P}$ appears on $\mathscr{E}$ for all points in a trapezoid of $\mathscr{M}^{\triangledown}$. For each trapezoid $\psi \in \mathscr{M}^{\triangledown}$, we construct the prism $\psi^{\downarrow} := \{(x, y, z) \in \mathbb{R}^3 \mid (x, y) \in \psi \text{ and } z \in (-\infty, \mathscr{E}(x, y)\}$; $\psi^{\downarrow}$ is unbounded in the $(-z)$-direction and bounded by the graph of $\mathscr{E}$ from above. It is easily seen that $\{\psi^{\downarrow} \mid \psi \in \mathscr{M}\}$ is a partition of $\mathscr{K}(\mathscr{P}^{\uparrow})$ into elementary cells. Furthermore, since $|\mathscr{M}^{\triangledown}| = O^*(m^2)$, the number of elementary cells in the partition is $O^*(m^2)$. Hence, we obtain the following:

▶ **Lemma 13.** *Let $P \subset \mathbb{R}^3$ be a set of $n$ points in $\mathbb{R}^3$, and let $r \geq 1$ be a parameter. A test set of size $O^*(r^4)$ for $\mathbb{C}_{\mathsf{u},\varphi}$ with respect to $P$ and $r$ that satisfies (C1)–(C4) with $\zeta(m) = O^*(m^2)$ can be computed in $O^*(r^4)$ time.*

## 5 The Dual Problem

In this section, we consider the *dual problem* mentioned in Section 1: Given a set $\mathscr{L}$ of $n$ lines in $\mathbb{R}^3$, preprocess $\mathscr{L}$ into a data structure that supports efficient unit-ball intersection detection (as well as reporting) queries. This problem can be formulated as a *point-enclosure* problem among a set of unit cylinders: Let $\mathscr{C}$ be the set of unit cylinders whose axes are the lines of $\mathscr{L}$. Preprocess $\mathscr{C}$ into a data structure that can quickly determine whether a query point $q \in \mathbb{R}^3$ lies in the union of the cylinders in $\mathscr{C}$, or report all such cylinders.

**Data structure.** We project the cylinders in $\mathscr{C}$ onto the $xy$-plane. (As in Section 2, we assume that none of the axes of the cylinders in $\mathscr{C}$ are parallel to the $yz$-plane.) Let $\mathscr{B}$ denote the set of boundary (silhouette) lines in $\mathbb{R}^2$ of the strips corresponding to the $xy$-projections of the cylinders in $\mathscr{C}$. Let $r > 1$ be a sufficiently large constant. We construct in $O(n^2)$ time a *hierarchical* $(1/r)$*-cutting* of $\mathscr{B}$ using the algorithm by Chazelle [6]. That is, we construct $s = O(\log n)$ cuttings $\Xi_1, \ldots, \Xi_s$ so that $\Xi_i$ is a $(1/r^i)$-cutting of $\mathscr{B}$ of size $O(r^{2i})$, each triangle of $\Xi_i$ is contained in a triangle of $\Xi_{i-1}$, and each triangle $\triangle \in \Xi_{i-1}$ contains a constant number of triangles of $\Xi_i$, which we refer to as *children* cells of $\triangle$. Each cell of $\Xi_s$ is crossed by $O(1)$ lines of $\mathscr{B}$. The algorithm also constructs the subset of lines of $\mathscr{B}$ crossing every cell of $\Xi_i$ for all $i \leq s$.

Fix a cell $\triangle$ of $\Xi_i$ for some $i \le s$. Let $\triangle^{\uparrow} := \triangle \times \mathbb{R}$ be the vertical slab erected over $\triangle$. Let $\mathscr{C}_\triangle \subseteq \mathscr{C}$ be the set of cylinders that intersect the slab $\triangle^{\uparrow}$. Following our definitions in Section 2, we call $\triangle$ (and $\triangle^{\uparrow}$) *narrow* if its width is at most 2 and *wide* otherwise. If a unit cylinder intersects a wide slab $\triangle^{\uparrow}$, then at least one of its two *silhouette* lines crosses $\triangle$. Hence, by the cutting property, for a wide cell $\triangle$ of $\Xi_i$, $|\mathscr{C}_\triangle| \le n/r^i$. If a cell $\triangle \in \Xi_i$ is narrow, then we construct a secondary data structure $\Psi_\triangle$ for answering point-enclosure queries on $\mathscr{C}_\triangle$, as described below. Furthermore, we remove all cells of $\Xi_j$, for $j > i$, that are contained in $\triangle$, for they will never be visited by the query procedure.

We now describe the secondary data structure constructed on a narrow cell $\triangle$. We assume that the width of $\triangle$ is at most $\sin^2(1/16)$, otherwise we split $\triangle$ into $O(1)$ subcells, each of width at most $\sin^2(1/16)$ and construct a separate secondary data structure for each of them. Let $\mathscr{Z}$ and $\Lambda(C)$, for a unit cylinder $C$, be the same as defined in Section 3. For each cylinder $C \in \mathscr{C}_\triangle$, we construct the collection $\Lambda(C)$ of canonical cylindrical prisms, as described in Section 3.1. Recall that each prism in $\Lambda(C)$ is erected in one of directions in $\mathscr{Z}$, i.e., it has the following form $\tau^{\uparrow}_{\mathsf{u}} = \hat{\tau} \times \mathsf{r}_{\mathsf{u}}$ where $\hat{\tau}$ is a clipped canonical patch of $C$. By Lemma 7, for a point $q \in \triangle^{\uparrow}$, $q \in C$ if and only if $q \in \mathscr{U}(\Lambda(C))$. We thus a build a data structure for answering point-enclosure queries in the set $\bigcup_{C \in \mathscr{C}_\triangle} \Lambda(C)$.

We fix a direction $\mathsf{u} \in \mathscr{Z}$ and let $\mathscr{P}^{\uparrow}_{\mathsf{u}} \subseteq \bigcup_{C \in \mathscr{C}_\triangle} \Lambda(C)$ be the subset of canonical prisms of cylinders in $\mathscr{C}_\triangle$ erected in direction $\mathsf{u}$. We build a separate data structure $\Psi_{\triangle,\mathsf{u}}$ for answering point-enclosure queries in $\mathscr{P}^{\uparrow}_{\triangle,\mathsf{u}}$, for every $\mathsf{u} \in \mathscr{Z}$, as follows. Without loss of generality, assume that $\mathsf{u}$ is the $(+z)$-direction. Let $\mathscr{P}_{\triangle,\mathsf{u}} = \{\hat{\tau}_{\mathsf{u}} \mid \tau^{\uparrow}_{\mathsf{u}} \in \mathscr{P}^{\uparrow}_{\triangle,\mathsf{u}}\}$ be the set of clipped canonical patches corresponding to the prisms in $\mathscr{P}^{\uparrow}_{\triangle,\mathsf{u}}$, which, as in Section 4, we regard as a set of partially-defined bivariate functions. Let $\mathscr{E}_{\triangle,\mathsf{u}}$ be the lower envelope of $\mathscr{P}_{\triangle,\mathsf{u}}$, and $\mathscr{M}_{\triangle,\mathsf{u}}$ its minimization diagram. Their complexity is $O^*(|\mathscr{P}_{\triangle,\mathsf{u}}|^2)$. A point $q = (q_x, q_y, q_z) \in \mathscr{U}(\mathscr{P}^{\uparrow}_{\triangle,\mathsf{u}})$ if and only if $q_z \ge \mathscr{E}_{\triangle,\mathsf{u}}(q_x, q_y)$. We construct $\mathscr{M}_{\triangle,\mathsf{u}}$ and preprocess it for answering planar point-location queries. Summing over all directions in $\mathscr{Z}$, the total size of the data structure $\Psi_\triangle$ is $O^*(|\mathscr{C}_\triangle|^2)$ and it can be constructed in time $O^*(|\mathscr{C}_\triangle|^2)$. Summing these bounds over all narrow cells of the hierarchical cuttings, the total size and the preprocessing time of the overall data structure are $O^*(n^2)$.

**Query procedure.** Let $q = (q_x, q_y, z_z)$ be a query point. We visit the cuttings $\Xi_1, \Xi_2, \ldots$ in order. Suppose we are visiting $\Xi_i$, and let $\triangle \in \Xi_i$ be the cell containing $q^* = (q_x, q_y)$. If $i = s$, we answer the query in $O(1)$ time by testing $q$ with all cylinders of $\mathscr{C}_\triangle$. If $\triangle$ is narrow, we query the secondary data structure $\Psi_\triangle$, as follows. For each direction $\mathsf{u} \in \mathscr{Z}$, we check whether $q \in \mathscr{U}(\mathscr{P}^{\uparrow}_{\mathsf{u}})$ by locating $(q_x, x_y)$ in $\mathscr{M}_{\mathsf{u}}$ and testing whether $q_z \ge \mathscr{E}_{\mathsf{u}}(q_x, q_y)$. If the answer is yes for one such $\mathsf{u}$, we conclude that $q \in \mathscr{U}(\mathscr{C}_\triangle)$ and return yes. Otherwise, we return no. Finally, if $\triangle$ is wide, we recursively visit the child cell of $\Xi_{i+1}$ that contains $q$. The overall query time is $O(\log n)$. This completes the proof of Theorem 2.

## References


1. Pankaj K. Agarwal. Range searching,. In J. E. Goodman, J. O'Rourke, and C. D. Tóth, editors, *Handbook of Discrete and Computational Geometry*, chapter 40, pages 1057–1092. Chapman and Hall/CRC, third edition, 2017. URL: `https://www.taylorfrancis.com/chapters/edit/10.1201/9781315119601-40/range-searching-pankaj-agarwal`.
2. Pankaj K. Agarwal, Boris Aronov, Esther Ezra, Matthew J. Katz, and Micha Sharir. Intersection queries for flat semi-algebraic objects in three dimensions and related problems. In *Proc. 38th Int. Sympos. Comput. Geom.*, volume 224 of *LIPIcs*, pages 4:1–4:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. `doi:10.4230/LIPIcs.SoCG.2022.4`.

3 Pankaj K. Agarwal and Je Erickson. Geometric range searching and its relatives. In B. Chazelle, J. E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, pages 1–56. Amer. Math. Soc., 2007.

4 Pankaj K. Agarwal and Jirí Matousek. On range searching with semialgebraic sets. *Discret. Comput. Geom.*, 11:393–418, 1994. doi:10.1007/BF02574015.

5 Timothy M. Chan. Optimal partition trees. *Discret. Comput. Geom.*, 47(4):661–690, 2012. doi:10.1007/s00454-012-9410-z.

6 Bernard Chazelle. Cutting hyperplanes for divide-and-conquer. *Discret. Comput. Geom.*, 9:145–158, 1993. doi:10.1007/BF02189314.

7 David P. Dobkin and Herbert Edelsbrunner. Space searching for intersecting objects. *J. Algorithms*, 8(3):348–361, 1987. doi:10.1016/0196-6774(87)90015-0.

8 Esther Ezra and Micha Sharir. On ray shooting for triangles in 3-space and related problems. *SIAM J. Comput.*, 51(4):1065–1095, 2022. doi:10.1137/21m1408245.

9 Larry Guth. Polynomial partitioning for a set of varieties. *Math. Proc. Camb. Phil. Soc.*, 159:459–469, 2015. URL: http://hdl.handle.net/1721.1/110285.

10 Vladlen Koltun. Almost tight upper bounds for vertical decompositions in four dimensions. *J. ACM*, 51(5):699–730, 2004. doi:10.1145/1017460.1017461.

11 Jirí Matousek. Reporting points in halfspaces. *Comput. Geom.*, 2:169–186, 1992. doi:10.1016/0925-7721(92)90006-E.

12 Shai Mohaban and Micha Sharir. Ray shooting amidst spheres in three dimensions and related problems. *SIAM J. Comput.*, 26(3):654–674, 1997. doi:10.1137/S0097539793252080.

13 Marco Pellegrini. Ray shooting on triangles in 3-space. *Algorithmica*, 9(5):471–494, 1993. doi:10.1007/BF01187036.

14 Marco Pellegrini. Ray shooting and lines in space. In Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Tóth, editors, *Handbook of Discrete and Computational Geometry*, pages 1093–1112. CRC Press, third edition edition, 2017.

15 Micha Sharir and Pankaj K. Agarwal. *Davenport-Schinzel sequences and their geometric applications.* Cambridge University Press, 1995.

16 Micha Sharir and Hayim Shaul. Semialgebraic range reporting and emptiness searching with applications. *SIAM J. Comput.*, 40(4):1045–1074, 2011. doi:10.1137/090765092.