SIAM J. COMPUT.
Vol. 47, No. 1, pp. 1–51

APPROXIMATE POLYTOPE MEMBERSHIP QUERIES[*]

SUNIL ARYA[†], GUILHERME D. DA FONSECA[‡], AND DAVID M. MOUNT[§]

Abstract. In the polytope membership problem, a convex polytope K in $\mathbb{R}^d$ is given, and the objective is to preprocess K into a data structure so that, given any query point $q \in \mathbb{R}^d$, it is possible to determine efficiently whether $q \in K$. We consider this problem in an approximate setting. Given an approximation parameter ε, the query can be answered either way if the distance from q to K's boundary is at most ε times K's diameter. We assume that the dimension d is fixed, and K is presented as the intersection of n halfspaces. Previous solutions to approximate polytope membership were based on straightforward applications of classic polytope approximation techniques by Dudley [*Approx. Theory*, 10 (1974), pp. 227–236] and Bentley, Faust, and Preparata [*Commun. ACM*, 25 (1982), pp. 64–68]. The former is optimal in the worst case with respect to space, and the latter is optimal with respect to query time. We present four main results. First, we show how to combine the two above techniques to obtain a simple space-time trade-off. Second, we present an algorithm that dramatically improves this trade-off. In particular, for any constant $\alpha \geq 4$, this data structure achieves query time roughly $O(1/\varepsilon^{(d-1)/\alpha})$ and space roughly $O(1/\varepsilon^{(d-1)(1-\Omega(\log \alpha)/\alpha)})$. We do not know whether this space bound is tight, but our third result shows that there is a convex body such that our algorithm achieves a space of at least $\Omega(1/\varepsilon^{(d-1)(1-O(\sqrt{\alpha})/\alpha})$. Our fourth result shows that it is possible to reduce approximate Euclidean nearest neighbor searching to approximate polytope membership queries. Combined with the above results, this provides significant improvements to the best known space-time trade-offs for approximate nearest neighbor searching in $\mathbb{R}^d$. For example, we show that it is possible to achieve a query time of roughly $O(\log n + 1/\varepsilon^{d/4})$ with space roughly $O(n/\varepsilon^{d/4})$, thus reducing by half the exponent in the space bound.

Key words. polytope membership, nearest neighbor searching, geometric retrieval, space-time trade-offs, approximation algorithms, convex approximation, Mahler volume

AMS subject classifications. 52B99, 68W25, 68P05

DOI. 10.1137/16M1061096

1. Introduction.

Convex polytopes are key structures in many areas of mathematics and computation. In this paper, we consider a fundamental search problem related to convex polytopes. Let K denote a convex body in $\mathbb{R}^d$, that is, a closed, convex set of bounded diameter that has a nonempty interior. We assume that K is presented as the intersection of n closed halfspaces. (Our results generally hold for any representation that satisfies the access primitives given at the start of section 3.) The *polytope membership problem* is that of preprocessing K so that it is possible to determine efficiently whether a given query point $q \in \mathbb{R}^d$ lies within K. Throughout, we assume that the dimension d is a fixed constant that is at least 2.

It follows from standard results in projective duality that polytope membership is equivalent to answering halfspace emptiness queries for a set of n points in $\mathbb{R}^d$. In dimension $d \leq 3$, it is possible to build a data structure of linear size that can answer such queries in logarithmic time [30]. In higher dimensions, however, all known exact

[*] Received by the editors February 11, 2016; accepted for publication (in revised form) June 2, 2017; published electronically January 2, 2018.
http://www.siam.org/journals/sicomp/47-1/M106109.html
Funding: The first author was supported by the Research Grants Council of Hong Kong, China, under projects 610108 and 16200014. The second author was supported by grants from CNPq and FAPERJ. The third author was supported by NSF grant CCF-1618866.

[†] Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (arya@cse.ust.hk).

[‡] Université Clermont Auvergne and LIMOS, Clermont-Ferrand, France (fonseca@isima.fr).

[§] Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742 (mount@umd.edu).

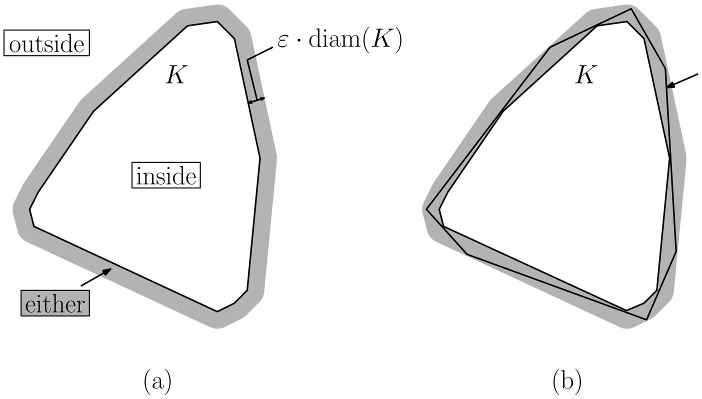


Fig. 1. *Approximate polytope membership:* (a) *problem formulation,* (b) *outer ε-approximation.*

data structures with roughly linear space have a query time of $\widetilde{\Omega}(n^{1-1/\lfloor d/2 \rfloor})$[1] [43], which is unacceptably high for many applications. Polytope membership is a special case of polytope intersection queries [25, 31, 13]. Barba and Langerman [13] showed that for any fixed d, it is possible to preprocess polytopes in $\mathbb{R}^d$ so that given two such polytopes that have been translated and rotated, it can be determined whether they intersect each other in time that is logarithmic in their total combinatorial complexity. However, the preprocessing time and space grow as the combinatorial complexity of the polytope raised to the power $\lfloor d/2 \rfloor$.

The lack of efficient exact solutions motivates the question of whether polytope membership queries can be answered approximately. Let ε be a positive real parameter, and let $\operatorname{diam}(K)$ denote K's diameter. Given a query point $q \in \mathbb{R}^d$, an *ε-approximate polytope membership query* returns a positive result if $q \in K$ and a negative result if the distance from q to its closest point in K is greater than $\varepsilon \cdot \operatorname{diam}(K)$, and it may return either result otherwise (see Figure 1(a)). Polytope membership queries, both exact and approximate, arise in many application areas, such as linear-programming and ray-shooting queries [20, 24, 46, 44, 42], nearest neighbor searching and the computation of extreme points [21, 28], collision detection [36], and machine learning [19].

Existing solutions to approximate polytope membership queries have been based on straightforward applications of classic polytope approximation techniques. We say that a polytope P is an *outer ε-approximation* of K if $K \subseteq P$, and the Hausdorff distance between P and K is at most $\varepsilon \cdot \operatorname{diam}(K)$ (see Figure 1(b)). An *inner ε-approximation* is defined similarly but with $P \subseteq K$. Dudley [32] showed that there exists an outer ε-approximating polytope for any bounded convex body in $\mathbb{R}^d$ formed by the intersection of $O(1/\varepsilon^{(d-1)/2})$ halfspaces, and Bronshteyn and Ivanov [17] proved an analogous bound on the number of vertices needed to obtain an inner ε-approximation. Both bounds are known to be asymptotically tight in the worst case (see, e.g., [18]). These results have been applied to a number of problems, for example, the construction of coresets [2]. By checking that a given query point lies within each of the halfspaces of Dudley's approximation, ε-approximate polytope membership queries can be answered with space and query time of $O(1/\varepsilon^{(d-1)/2})$.

[1] Throughout, we use $\widetilde{O}(\cdot)$ and $\widetilde{\Omega}(\cdot)$ as variants of $O(\cdot)$ and $\Omega(\cdot)$, respectively, that ignore logarithmic factors. We use "lg" to denote base-2 logarithm.

The principal contribution of this paper is to show that it is possible to achieve nontrivial space-time trade-offs for approximate polytope membership. In order to motivate our methods, in section 2 we present a simple space-time trade-off (stated in the following theorem), which is based on a straightforward combination of the approximations of Dudley [32] and Bentley, Faust, and Preparata [15]. Throughout, we will treat n and ε as asymptotic quantities, while the dimension d is a constant.

THEOREM 1.1 (simple trade-off). *Given a convex polytope K in $\mathbb{R}^d$, a positive approximation parameter ε, and a real parameter $\alpha \geq 2$, there is a data structure for ε-approximate polytope membership queries that achieves*

$$\textit{Query time: } O\left(1/\varepsilon^{\frac{d-1}{\alpha}}\right) \qquad \textit{Space: } O\left(1/\varepsilon^{(d-1)\left(1-\frac{1}{\alpha}\right)}\right).$$

The constant factors in the space and query time depend only on d (not on K, α, or ε).

We will strengthen this trade-off significantly in sections 3 and 4. We will show that it is possible to build a data structure with $O(1/\varepsilon^{(d-1)/2})$ space that allows polytope membership queries to be answered in roughly $O(1/\varepsilon^{(d-1)/8})$ time, thus reducing the exponent in the query time of Theorem 1.1 (for $\alpha = 2$) by 1/4. Further, we will show that by iterating a suitable generalization of this construction, we can obtain the following trade-offs.

THEOREM 1.2. *Given a convex polytope K in $\mathbb{R}^d$, an approximation parameter $0 < \varepsilon \leq 1$, and a real constant $\alpha \geq 4$, there is a data structure for ε-approximate polytope membership queries that achieves*

$$\textit{Query time: } O\left(\left(\log \frac{1}{\varepsilon}\right)/\varepsilon^{\frac{d-1}{\alpha}}\right) \qquad \textit{Space: } O\Big(1/\varepsilon^{(d-1)\left(1-\frac{2\lfloor \lg \alpha \rfloor - 2}{\alpha}\right)}\Big).$$

The constant factors in the space and query time depend only on d and α (not on K or ε).

The above space bound is a simplification, and the exact bound is given in Lemma 6.4. Both bounds are piecewise linear in $1/\alpha$ (with breakpoints at powers of two), but the bounds of Lemma 6.4 are continuous as a function of α. The resulting space-time trade-off is illustrated in Figure 2(a). (The plot reflects the more accurate bounds.)

The above theorem is intentionally presented in a purely existential form. This is because our construction algorithm assumes the existence of a procedure that computes an ε-approximating polytope whose number of bounding hyperplanes is at most a constant factor larger than optimal. Unfortunately, we know of no efficient solution to this problem. In Lemma 7.6 we will show that if the input polytope is expressed as the intersection of n halfspaces, it is possible to build such a structure in time $O(n + 1/\varepsilon^{O(1)})$, such that the space and query times of the above theorem increase by an additional factor of $O(\log \frac{1}{\varepsilon})$.

Note that in contrast to many complexity bounds in the area of convex approximation, which hold only in the limit as ε approaches zero (see, e.g., [37, 40]), Theorems 1.1 and 1.2 hold for any positive $\varepsilon \leq 1$. The data structure of Theorem 1.2 is quite simple. It is based on a quadtree subdivision of space in which each cell is repeatedly subdivided until the combinatorial complexity of the approximating polytope within the cell is small enough to achieve the desired query time.

We do not know whether the upper bounds presented in Theorem 1.2 are tight for our algorithm. In section 8, we establish the following lower bound on the trade-off achieved by this algorithm.

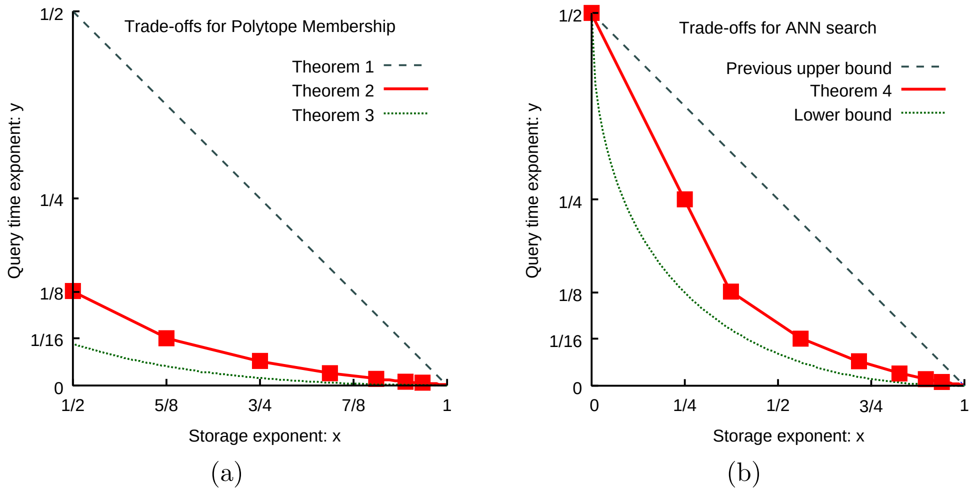


Fig. 2. *The multiplicative factors in the exponent of the* $1/\varepsilon$ *terms for* (a) *polytope membership queries and* (b) *approximate nearest neighbor (ANN) queries. Each point* (x, y) *represents a term of* $1/\varepsilon^{x(d \pm O(1))}$ *for storage and* $1/\varepsilon^{y(d \pm O(1))}$ *for query time, where the* $O(1)$ *term does not depend on* d.

Theorem 1.3. *In any fixed dimension* $d \geq 2$ *and for any constant* $\alpha \geq 4$*, there exists a polytope such that for all sufficiently small positive* ε*, the data structure described in Theorem* 1.2 *when generated to achieve query time* $O(1/\varepsilon^{(d-1)/\alpha})$ *has space*

$$\Omega\left(1/\varepsilon^{(d-1)\left(1-\frac{2\sqrt{2\alpha}-3}{\alpha}\right)-1}\right).$$

Although α is not an asymptotic quantity, for the sake of comparing the upper and lower bounds, let us imagine that it is. For roughly the same query time, the α dependencies appearing in the exponents of the upper bounds on space are $(1-\frac{1}{\alpha})$ for Theorem 1.1 and $(1-\frac{\Omega(\log \alpha)}{\alpha})$ for Theorem 1.2, and the lower bound of Theorem 1.3 is roughly $(1-\frac{O(\sqrt{\alpha})}{\alpha})$. The trade-offs provided in these theorems are illustrated in Figure 2(a).

The second major contribution of this paper is to demonstrate that our trade-offs for approximate polytope membership queries imply significant improvements to the best known space-time trade-offs for approximate nearest neighbor searching. We are given a set X of n points in $\mathbb{R}^d$. Given any $q \in \mathbb{R}^d$, an ε-*approximate nearest neighbor* of q is any point of X whose distance from q is at most $(1+\varepsilon)$ times the distance to q's closest point in X. The objective is to preprocess X in order to answer such queries efficiently. Data structures for approximate nearest neighbor searching (in fixed dimensions) have been proposed by several authors [22, 33, 38, 28, 47]. The best space-time trade-offs [10] have query times roughly $O(1/\varepsilon^{d/\alpha})$ with storage roughly $O(n/\varepsilon^{d(1-2/\alpha)})$ for $\alpha \geq 2$ (see the dashed line in Figure 2(b)).

These results are based on a data structure called an *approximate Voronoi diagram* (AVD). In general, a data structure for approximate nearest neighbor searching is said to be in the *AVD model* if it has the general form of decomposition of space (generally a covering) by hyperrectangles of bounded aspect ratio, each of which is associated with a set of representative points. Given any hyperrectangle that contains the query point, at least one of these representatives is an ε-approximate nearest neighbor of the query point [10]. The AVD model is of interest because it is possible to prove lower bounds on the performance of such a data structure. In particular, the lower bounds proved in [10] are shown in the dotted curve in Figure 2(b). By violating the AVD model, small additional improvements were obtained in [6].

Our improvements to approximate nearest neighbor searching are given in the following theorem.

THEOREM 1.4. *Let $0 < \varepsilon \leq 1$ be a real parameter, $\alpha \geq 1$ be a real constant, and X be a set of n points in $\mathbb{R}^d$. There is a data structure in the AVD model for approximate nearest neighbor searching that achieves*

$$\textit{Query time: } O\biggl(\log n + (1/\varepsilon^{d/2\alpha}) \cdot \log^2 \frac{1}{\varepsilon}\biggr),$$

$$\textit{Space: } O\biggl(n \cdot \max\biggl(\log \frac{1}{\varepsilon}, 1/\varepsilon^{d\left(\frac{1}{2} - \frac{1}{2\alpha}\right)}\biggr)\biggr) \textit{ for } 1 \leq \alpha < 2, \textit{ and}$$

$$O\Bigl(n/\varepsilon^{d\left(1 - \frac{\lfloor \lg \alpha \rfloor}{\alpha} - \frac{1}{2\alpha}\right)}\Bigr) \textit{ for } \alpha \geq 2.$$

The constant factors in the space and query time depend only on d and α (not on ε).

The above space bound is a simplification of the more accurate bound given in Lemma 9.5. (Also see the remarks following the proof of this lemma for further minor improvements achievable by forgoing the AVD model.) As before, both bounds are piecewise linear in $1/\alpha$ (with breakpoints at powers of two), but the bounds of Lemma 9.5 are continuous as a function of α. The resulting space-time trade-off is illustrated in Figure 2(b). (The plot reflects the more accurate bounds of Lemma 9.5.)

As an example of the strength of the improvement that this offers, observe that in order for the existing AVD-based results to yield a query time of $\widetilde{O}(1/\varepsilon^{d/4})$ the required space would be roughly $\Omega(n/\varepsilon^{d/2})$. The exponent in the space bound is nearly twice that given by Theorem 1.4, which arises by setting $\alpha = 2$. The connection between the polytope membership problem and approximate nearest neighbor searching has been noted before by Clarkson [28]. Unlike Clarkson's, our results hold for point sets with arbitrary aspect ratios.

Our data structure is based on a simple quadtree-based decomposition of space. Let t denote the desired query time. We begin by preconditioning K so that it is fat and has at most unit diameter. We then employ a quadtree that hierarchically subdivides space into hypercube cells. The decomposition stops whenever we can declare that a cell is either entirely inside or outside of K, or (if it intersects K's boundary) it is locally approximable by at most t halfspaces. This procedure, called SplitReduce, is presented in section 3. Queries are answered by descending the quadtree to determine the leaf cell containing the query point, and (if not inside or outside) testing whether the query point lies within the approximating halfspaces.

Although the algorithm itself is very simple, the analysis of its space requirements is quite involved. In section 4, we begin with a simple analysis, which shows that it is possible to obtain a significant improvement over the Dudley-based approach (in particular, reducing the exponent in the query time by half with no increase in space). While this simple analysis introduces a number of useful ideas, it is not tight nor does it provide space-time trade-offs.

Our final analysis requires a deeper understanding of the local structure of the convex body's boundary. In section 5 we introduce local surface patches of K's boundary, called ε-dual caps. We relate the data structure's space requirements to the existence of a low cardinality hitting set of the dual caps. We present a two-pronged strategy for generating such a hitting set, one focused on dual caps of large surface area (intuitively corresponding to boundary regions of low curvature) and the other focused on dual caps of small surface area (corresponding to boundary regions of high curvature). We show that simple random sampling suffices to hit dual caps of high surface area, and so the challenge is to hit the dual caps of low surface area. To do this, we show that dual caps of low surface area generate Voronoi patches on a hypersphere enclosing K of large surface area. We refer to this result as the *area-product bound*, which is stated in Lemma 5.2. This admits a strategy based on sampling points randomly on this hypersphere, and then projecting them back to their nearest neighbor on the surface of K.

The area-product bound is proved with the aid of a classical concept from the theory of convexity, called the *Mahler volume* [16, 48]. The Mahler volume of a convex body is a dimensionless quantity that involves the product of the body's volume and the volume of its polar body. We demonstrate that dual caps and their Voronoi patches exhibit a similar polar relationship. The proof of the area-product bound is quite technical and is deferred to section 10.

Armed with the area-product bound, in section 6 we establish our final bound on the space-time trade-offs of SplitReduce, which culminates in the proof of Theorem 1.2. In section 7 we present details on how the data structure is built and discuss preprocessing time. In section 8 we establish the lower bound result, which is stated in Theorem 1.3.

Finally, in section 9 we show how these results can be applied to improve the performance of approximate nearest neighbor searching in Euclidean space. It is well known that (exact) nearest neighbor searching can be reduced to vertical ray shooting to a polyhedron that results by lifting points in dimension d to tangent hyperplanes for a paraboloid in dimension $d+1$ [3, 34]. We show how to combine approximate vertical ray shooting (based on approximate polytope membership) with AVDs to establish Theorem 1.4.

2. Preliminaries.

Throughout, we will use asymptotic notation to eliminate constant factors. In particular, for any positive real x, let $O(x)$ denote a quantity that is at most cx for some constant c. Define $\Omega(x)$ and $\Theta(x)$ analogously. We will sometimes introduce constants within a local context (e.g., within the statement of single lemma). To simplify notation, we will often use the same symbol "c" to denote such generic constants. Recall that we use "lg" to denote the base-2 logarithm. We will use "log" when the base does not matter. Some of our search algorithms involve integer grids, and for these we assume a model of computation that supports integer division.

Let K denote a full-dimensional convex body in $\mathbb{R}^d$, and let ∂K denote its boundary. For concreteness, we assume that K is represented as the intersection of n closed halfspaces. Our data structure can generally be applied to any representation that supports access primitives (i)–(iii) given at the start of section 3.

2.1. Absolute and relative approximations.

Earlier, we defined approximation relative to K's diameter, but it will be convenient to define the approximation error in absolute terms. Given a positive real r, define $K \oplus r$ to be a set of points that lie within Euclidean distance r of K. We say that a polytope P is an *absolute ε-approximation* of a convex body K if

$$K \subseteq P \subseteq K \oplus \varepsilon.$$

When we wish to make the distinction clear, we refer to the definition in the introduction as a *relative approximation*. Henceforth, unless otherwise stated approximations are in the absolute sense.

In order to reduce the general approximation problem into a more convenient absolute form, we will transform K into a "fattened" body of bounded diameter. Given a parameter $0 < \gamma \leq 1$, we say that a convex body K is *γ-fat* if there exist concentric Euclidean balls B and B', such that $B \subseteq K \subseteq B'$, and $\text{radius}(B)/\text{radius}(B') \geq \gamma$. We say that K is *fat* if it is γ-fat for a constant γ (possibly depending on d, but not on n or ε). The following lemma shows that K can be fattened without significantly altering the approximation parameter. Let $Q_0^{(d)}$ denote the d-dimensional axis-aligned hypercube of unit diameter centered at the origin. When d is clear, we refer to this as Q_0.

LEMMA 2.1. *Given a convex body K in $\mathbb{R}^d$ and $0 < \varepsilon \leq 1$, there exists an affine transformation T such that $T(K)$ is $(1/d)$-fat and $T(K) \subseteq Q_0$. If P is an absolute $(\varepsilon/d\sqrt{d})$-approximation of $T(K)$, then $T^{-1}(P)$ is a relative ε-approximation of K.*

We omit the proof of this lemma for now, since it is subsumed by Lemma 7.1 below. Our approach will be to map K to $T(K)$, set $\varepsilon' \leftarrow \varepsilon/d\sqrt{d}$, and then apply an absolute ε'-approximation algorithm to $T(K)$ (or more accurately, to the result of applying T to each of K's defining halfspaces). Since ε' is within a constant factor of ε, the asymptotic complexity bounds that we will prove for the absolute case will apply to the original (relative) approximation problem case as well.

2.2. Concepts from quadtrees. By the above reduction, it suffices to consider the problem of computing an absolute ε-approximation to a fat convex body K that lies within Q_0. Our construction will be based on a quadtree decomposition of Q_0. More formally, we define a *quadtree cell* by the following well-known recursive decomposition. Q_0 is a quadtree cell, and given any quadtree cell Q, each of the 2^d hypercubes that result by bisecting each of Q's sides by an axis-orthogonal hyperplane is also a quadtree cell. A cell Q' that results from subdividing Q is a *child* of Q. Clearly, the child's diameter is half that of its parent. The subdivision process defines a (2^d)-ary tree whose nodes are quadtree cells and whose leaves are cells that are not subdivided.

It will be useful to define a notion of approximation that is local to a quadtree cell Q. An obvious definition would be to approximate $K \cap Q$. The problem with this is that a point $p \in Q$ that is close to K need not be close to $K \cap Q$ (see Figure 3(a)). To remedy this we say that a polytope P is an *ε-approximation of K within Q* if

$$K \cap Q \ \subseteq \ P \cap Q \ \subseteq \ (K \oplus \varepsilon) \cap Q$$

(see Figure 3(b)). This definition implies that for any query point $q \in Q$, we can correctly answer ε-approximate polytope membership queries with respect to K by checking whether $q \in P$. We do not care what happens outside of Q, and indeed P may even be unbounded.

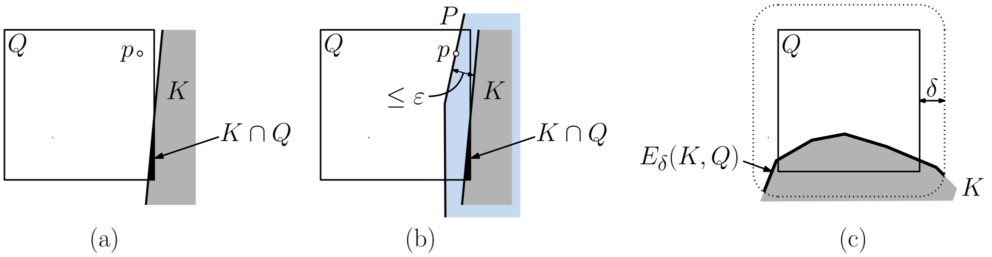


FIG. 3. (a), (b) *ε-approximation of K within Q, and* (c) $E_\delta(K,Q)$.

As we shall see later, computing an ε-approximation of K within a quadtree cell Q will generally require that we consider ∂K in a region that extends slightly beyond Q. We define $E_\delta(K, Q)$ to be the portion of ∂K that lies within distance δ of Q (see Figure 3(c)). Because $\delta = \sqrt{\varepsilon}$ will be of particular interest, we use $E(K, Q)$ as shorthand for $E_{\sqrt{\varepsilon}}(K, Q)$.

In order to apply constructions on quadtree cells of various sizes, it will be convenient to transform all such constructions into a common form. Given a quadtree cell Q, we define *standardization* to be the application of an affine transformation that uniformly scales and translates space so that Q is aligned with the standard quadtree cell Q_0. We transform K using this same transformation and apply the same scale factor to ε. Although we assume that the input body is contained within Q_0, after standardization, the transformed image of K need not be contained within Q_0.

2.3. Polarity and the Mahler volume. Some of our analysis will involve the well-known concept of *polarity*. Let us recall some general facts (see, e.g., Eggleston [35]). Given vectors $u, v \in \mathbb{R}^d$, let $\langle u, v\rangle$ denote their inner product, and let $\|v\| = \sqrt{\langle v, v\rangle}$ denote v's Euclidean length. Given a convex body $K \in \mathbb{R}^d$ define its *polar* to be the convex set

$$\operatorname{polar}(K) \;=\; \{u : \langle u, v\rangle \le 1 \text{ for all } v \in K\}.$$

If K contains the origin, then $\operatorname{polar}(K)$ is bounded. Given $v \in \mathbb{R}^d$, $\operatorname{polar}(v)$ is simply the closed halfspace that contains the origin whose bounding hyperplane is orthogonal to v and at distance $1/\|v\|$ from the origin (on the same side of the origin as v). The polar has the inclusion-reversing property that v lies within $\operatorname{polar}(u)$ if and only if u lies within $\operatorname{polar}(v)$. We may equivalently define $\operatorname{polar}(K)$ as the intersection of $\operatorname{polar}(v)$ for all $v \in K$.

Generally, given $r > 0$, define

$$\operatorname{polar}_r(K) \;=\; \{u : \langle u, v\rangle \le r^2 \text{ for all } v \in K\}.$$

It is easy to see that for any $v \in \mathbb{R}^d$, $\operatorname{polar}_r(v)$ is the closed halfspace at distance $r^2/\|v\|$ (see Figure 4(a)). Thus, $\operatorname{polar}_r(K)$ is a uniform scaling of $\operatorname{polar}(K)$ by a factor of r^2. In particular, if B is a Euclidean ball of radius x centered at the origin, then $\operatorname{polar}_r(B)$ is a concentric ball of radius r^2/x.

An important concept related to polarity is the *Mahler volume*, which is defined to be the product of the volumes of a convex body and its polar. There is a large literature on the Mahler volume, mostly for centrally symmetric bodies. Later in the paper we will make use of the following bound on the Mahler volume for arbitrary

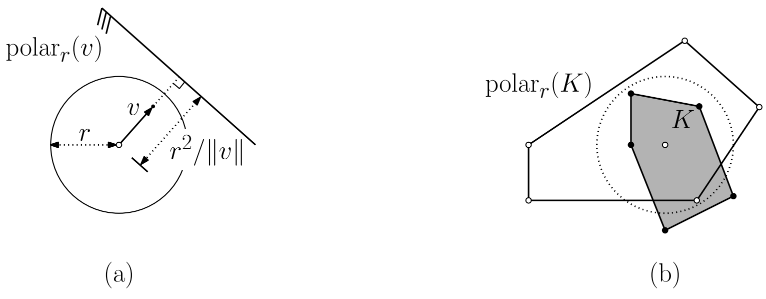


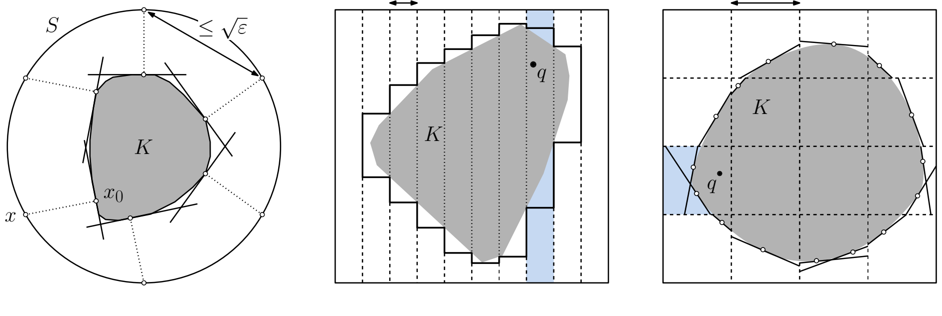
FIG. 4. *The generalized polar transform and polar body.*

convex bodies (see, e.g., Kuperberg [41]). Given a convex body K in $\mathbb{R}^d$, let $\mathrm{vol}(K)$ denote its volume or, more formally, its d-dimensional Hausdorff measure.

Lemma 2.2 (Mahler volume). *There is a constant c_m depending only on d such that given a convex body K in $\mathbb{R}^d$, $\mathrm{vol}(K) \cdot \mathrm{vol}(\mathrm{polar}(K)) \geq c_m$. More generally, given $r > 0$, $\mathrm{vol}(K) \cdot \mathrm{vol}(\mathrm{polar}_r(K)) \geq c_m\, r^{2d}$.*

2.4. Simple approximation trade-off. Before presenting our results, it will be illuminating to see how to obtain simple data structures for approximate polytope membership by combining two existing approximation methods. Let us begin by describing Dudley's approximation. Assuming that K is contained within Q_0, let S denote the $(d-1)$-dimensional sphere of radius 3 centered at the origin, which we call the *Dudley hypersphere*. (The value 3 is not critical; any sufficiently large constant suffices.) For $\delta > 0$, a set Σ of points on S is said to be δ-*dense* if every point of S lies within distance δ of some point of Σ. Let Σ be a $\sqrt{\varepsilon}$-dense set of points on S (see Figure 5(a)). By a simple packing argument there exists such a set of cardinality $\Theta(1/\varepsilon^{(d-1)/2})$. For each point $x \in \Sigma$, let x_0 be its nearest point on K's boundary. For each such point x_0, consider the halfspace containing K that is defined by the supporting hyperplane passing through x_0 that is orthogonal to the line segment $\overline{xx_0}$. Dudley shows that the intersection of these halfspaces is an outer ε-approximation of K. We can answer approximate membership queries by testing whether q lies within all these halfspaces (by brute force). This approach takes $O(1/\varepsilon^{(d-1)/2})$ query time and space.

An alternative solution is related to a grid-based approximation by Bentley, Faust, and Preparata [15]. Again, we assume that K is contained within Q_0. For the sake of illustration, let us think of the dth coordinate axis as pointing upward. Partition the upper facet of Q_0 into a $(d-1)$-dimensional square grid with cells of diameter ε. A packing argument implies that the number of cells is $O(1/\varepsilon^{d-1})$. Extend each of these cells downward to form a subdivision of Q_0 into vertical columns (see Figure 5(b)). Trim each column at the highest and lowest points at which it intersects K. Together, these trimmed columns define a collection of hyperrectangles whose union contains K. The resulting data structure has $O(1/\varepsilon^{d-1})$ space. Given a query point q, in $O(1)$ time we can determine the vertical column containing q (assuming a model of computation that supports integer division), and we then test whether q lies within the trimmed column. In contrast to the method based on Dudley's construction, this method provides a better query time of $O(1)$ but with higher space of $O(1/\varepsilon^{d-1})$.



(a) (b) (c)

Fig. 5. *The ε-approximations of* (a) *Dudley and* (b) *Bentley et al., and* (c) *the simple trade-off. (Not drawn to scale.)*

It is possible to combine these two solutions into a simple unified approach that achieves a trade-off between space and query time. Given a parameter r, where $\varepsilon \leq r \leq 1$, subdivide Q_0 into a grid of hypercube cells each of diameter $\Theta(r)$. For each cell Q that intersects K's boundary, apply Dudley's approximation to this portion of the polytope. By a straightforward packing argument, the number of grid cells that intersect K's boundary is $O(1/r^{d-1})$ (see, for example, Lemma 3 of [11]). We apply standardization to Q (thus mapping Q to Q_0 and scaling ε to $\Omega(\varepsilon/r)$) and apply Dudley's construction. By Dudley's results, the number of halfspaces needed to approximate K within Q is $O((r/\varepsilon)^{(d-1)/2})$. To answer a query, in $O(1)$ time we determine which hypercube of the grid contains the query point (assuming a model of computation that supports integer division). We then apply brute-force search to determine whether the query point lies within all the associated halfspaces in time $O((r/\varepsilon)^{(d-1)/2})$. The query time is dominated by this latter term. The space is dominated by the total number of halfspaces, which is $O((1/r^{d-1}) \cdot (r/\varepsilon)^{(d-1)/2}) = O(1/(\varepsilon r)^{(d-1)/2})$. If we express r in terms of a parameter α, where $r = \varepsilon^{1-2/\alpha}$, then Theorem 1.1 follows as an immediate consequence. Note that the resulting trade-off interpolates nicely between the two extremes for $\varepsilon \leq r \leq 1$.

3. The data structure and construction.

In this section we show how to improve the approach from the previous section by replacing the grid with a quadtree. The data structure is constructed by the recursive procedure, called SplitReduce, whose inputs consist of a convex body K and a quadtree cell Q. We are also given the approximation parameter $0 < \varepsilon \leq 1$ and a parameter $t \geq 1$ that controls the query time. Although we assume that K is presented as the intersection of n halfspaces, this procedure can be applied to any representation that supports the following access primitives:

(i) Determine whether Q is disjoint from K.
(ii) Determine whether Q is contained within $K \oplus \varepsilon$.
(iii) Determine whether there exists a set of at most t halfspaces whose intersection ε-approximates K within Q, and if so generate such a set.

Recall that we assume that K has been transformed so it is $(1/d)$-fat and lies within Q_0 (the hypercube of unit diameter centered at the origin). The data structure is built by the call SplitReduce(K, Q_0). In general, SplitReduce(K, Q) checks whether any of the above access primitives returns a positive result, and if so it terminates the decomposition and Q is declared a *leaf cell*. Otherwise, it makes a recursive call on the children of Q (see Figure 6(a)). On termination, each leaf cell is labeled as either "inside" or "outside" or is associated with a set of at most t approximating halfspaces (see Figure 6(b)).

```
SplitReduce(K, Q).
1. If $Q \cap K = \emptyset$, label $Q$ as "outside."
2. If $Q \subseteq K \oplus \varepsilon$, label $Q$ as "inside."
3. If there exists a set at most $t$ halfspaces whose intersection provides an
   $\varepsilon$-approximation to $K$ within $Q$, associate $Q$ with such a set $P(Q)$ of minimum size.
4. Otherwise, split $Q$ into $2^d$ quadtree cells and recursively invoke SplitReduce
   on each.
```

For the sake of our space-time trade-offs, we will usually assume that t is reasonably large, say, $t = \Omega(\log \frac{1}{\varepsilon})$. Under our assumption that $t \geq 1$, steps 1 and 2 are not needed, since it is possible to ε-approximate any cell satisfying these conditions with a single halfspace. This assumption on the value of t is mainly a convenience to simplify

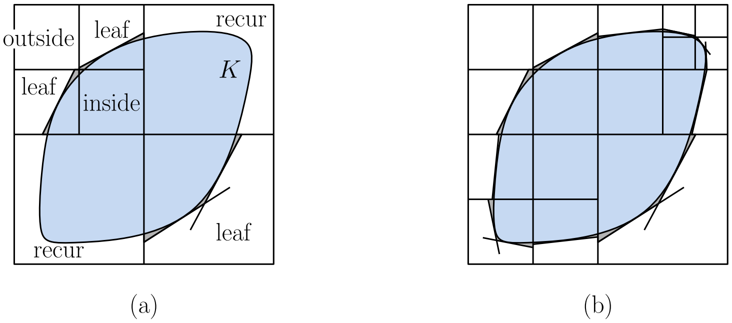


FIG. 6. (a) *Cases arising in SplitReduce for* $t = 2$ *and* (b) *the final subdivision.*

the formulas of our mathematical analysis. Observe that even if $t = 0$, the procedure will terminate and provide a correct answer once the cell diameter falls below ε.

It is easy to see that the recursion must terminate as soon as $\mathrm{diam}(Q) \leq \varepsilon$ (since, irrespective of whether it intersects ∂K, any such cell can be labeled either as "inside" or "outside"). Of course, it may terminate much sooner. Since Q_0 is of unit diameter, it follows that the height of the quadtree is $O(\log \frac{1}{\varepsilon})$. The total space used by the data structure is the sum of the space needed to store the quadtree and the space needed to store the approximating halfspaces for the cells that intersect K's boundary. Our next lemma shows that if the query time is sufficiently large, the latter quantity dominates the space asymptotically. For each leaf cell Q generated by step 3, define $t(Q) = |P(Q)|$, and define $t(Q) = 1$ for all the other leaf cells.

LEMMA 3.1. *Given a convex body* $K \subseteq Q_0$ *in* $\mathbb{R}^d$. *If* t *is* $\Omega(\log \frac{1}{\varepsilon})$, *then the total space of the data structure produced by SplitReduce*(K, Q_0) *on* K *for query time* t *is asymptotically dominated by the sum of* $t(Q)$ *over all the leaf cells* Q *that intersect* K*'s boundary.*

Proof. Let T denote the quadtree produced by running SplitReduce on K. As mentioned above, T is of height $O(\log \frac{1}{\varepsilon})$. By our hypothesis that t is $\Omega(\log \frac{1}{\varepsilon})$, there exists a constant c such that $\mathrm{height}(T) \leq ct$. Let L denote the set of leaves of T that intersect the boundary of K, and let M denote the internal nodes of T that have the property that all their children are leaves. (These are the lowest internal nodes of the tree.) Let $t(L)$ denote the sum of $t(Q)$ over all $Q \in L$.

The fact that each node $u \in M$ was subdivided by SplitReduce implies that more than t halfspaces are needed to approximate K within u's cell. Therefore, the children of u that intersect K's boundary together require at least t halfspaces. In addition to $P(Q)$, each quadtree leaf Q can (implicitly) contribute its $2d$ bounding hyperplanes to the approximation. Therefore, $t(L) + 2d|L|$ hyperplanes suffice to approximate K in all the cells of M, implying that $t(L) + 2d|L| \geq t \cdot |M|$. Since $t(Q) \geq 1$, we have $t(L) \geq |L|$, and thus $(1 + 2d)t(L) \geq t \cdot |M|$.

Each internal node of T either is in M or is an ancestor of a node in M. Thus, the total number of internal nodes of T is at most $|M| \cdot \mathrm{height}(T)$. Since each internal node of a quadtree has 2^d children, the total number of nodes in the tree, excluding the root, is at most

$$2^d \cdot |M| \cdot \mathrm{height}(T) \;\leq\; 2^d \cdot |M| \cdot (ct) \;\leq\; 2^d c(1 + 2d)t(L) \;=\; O(t(L)).$$

Each internal node of T and each leaf node that does not intersect K's boundary contributes only a constant amount to the total space. Therefore, the space contribution of the nodes other than those of L is at most a constant factor larger than the total number of nodes of T, which we have shown is $O(t(L))$. Therefore, the total space is $O(t(L))$, as desired. ∎

A query is answered by performing a point location in the quadtree to determine the leaf cell containing the query point. If the leaf cell is not labeled as being "inside" or "outside," we test whether the query point lies within all the associated halfspaces, and if so, we declare the point to be inside K. Otherwise it is declared to be outside. Clearly, the query time is $O(\log \frac{1}{\varepsilon} + t)$.

The algorithm is correct provided that the set of halfspaces $P(Q)$ computed in step 3 defines any ε-approximation of K within Q, but our analysis of the data structure's space requirements below (see the proof of Lemma 4.3) relies on the assumption that the size of $P(Q)$ is within a constant factor of the minimum number of halfspaces of any ε-approximating polytope within Q. Unfortunately, we know of no constant-factor approximation to the problem of computing such a polytope. Thus, strictly speaking, the bounds stated in Theorem 1.2 are purely existential. In section 7 we will show that through a straightforward modification of the greedy set-cover heuristic, it is possible to compute an approximation in which the number of defining halfspaces exceeds the optimum (for slightly smaller approximation parameter) by a factor of at most $\rho = O(\log \frac{1}{\varepsilon})$. From the following result it follows that this increases our space and query time bounds by $O(\log \frac{1}{\varepsilon})$.

Lemma 3.2. *Given any $\rho \geq 1$ and any constant $0 < \beta \leq 1$, if the number of halfspaces of $P(Q)$ computed in step* 3 *of SplitReduce is within a factor ρ of the minimum number of facets of any $(\beta\varepsilon)$-approximating polytope within Q, then Theorems* 1.2 *and* 1.4 *hold but with the asymptotic space and query time bounds larger by a factor of ρ.*

Proof. Let us refer to the hypothesized version of SplitReduce whose step 3 is suboptimal as SplitReduce′. Consider an execution of SplitReduce′ using ρt as the desired query time and ε as the approximation parameter, and let us compare this to an execution of SplitReduce using t and $\beta\varepsilon$, respectively. Since β is a constant, the asymptotic dependencies on ε are unaffected, and therefore the space and query times stated in Theorems 1.2 and 1.4 apply without modification to the execution of SplitReduce. In this execution, if the subdivision declares some quadtree cell Q to be a leaf, then t halfspaces suffice to $(\beta\varepsilon)$-approximate K within Q, and so by our hypothesis in the corresponding execution of SplitReduce′, step 3 returns at most ρt halfspaces, implying this execution also declares Q to be a leaf. Therefore, the tree generated by SplitReduce′ is a subtree of the tree generated by SplitReduce, but each leaf node may contain up to a factor of ρ more halfspaces. Thus, the asymptotic space and query time bounds for SplitReduce′ are larger than those of SplitReduce by this same factor. ∎

4. Simple upper bound. In this section, we present a simple upper bound of $O(1/\varepsilon^{(d-1)/2})$ on the storage of the data structure obtained by the SplitReduce algorithm for any given query time $t \geq 1/\varepsilon^{(d-1)/4}$. The tools developed in this section will be useful for the more comprehensive upper bounds, which will be presented in subsequent sections.

Throughout this section we do not necessarily assume that K has been scaled to lie within Q_0 and may generally be much larger. Recall that S denotes a hypersphere of radius 3 centered at the origin. Let X denote a surface patch of K that lies within S. Let $\mathrm{Vor}(X)$ denote the set of points exterior to K whose closest point on ∂K lies

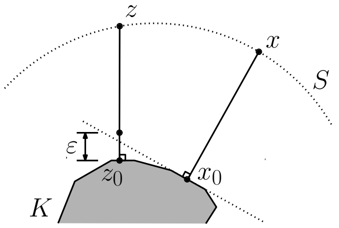


FIG. 7. *Lemma* 4.2.

within X. We refer to the surface patch $\mathrm{Vor}(X) \cap S$ (the points of S whose closest point on ∂K lies within X) as the *Voronoi patch* of X. Voronoi patches are related to Dudley's construction. In particular, a sample point $x \in S$ from Dudley's construction generates a supporting halfspace at a point of X if and only if $x \in \mathrm{Vor}(X) \cap S$. The following two lemmas are straightforward adaptations of Dudley's analysis [32]. The first is just a restatement of Dudley's result.

LEMMA 4.1. *Given a convex body K in $\mathbb{R}^d$ that lies within Q_0 and $0 < \varepsilon \leq 1$, there exists an ε-approximating polytope P bounded by at most $c/\varepsilon^{(d-1)/2}$ facets, where c is a constant depending only on d.*

The second lemma is a technical result that is implicit in Dudley's analysis. Given two points $x, y \in \mathbb{R}^d$, let $\overline{xy}$ denote the segment between them, and let $\|xy\|$ denote the Euclidean length of this segment.

LEMMA 4.2. *Let K be a convex body, let $0 < \varepsilon \leq 1$, and let z and x be two points of S such that $\|zx\| \leq \sqrt{\varepsilon}/4$. Let z_0 and x_0 be the points on ∂K that are closest to z and x, respectively. If z_0 is within unit distance of the origin, then*

(i) *$\|z_0x_0\| \leq \sqrt{\varepsilon}/4$ and*

(ii) *the supporting hyperplane at x_0 orthogonal to the segment $\overline{xx_0}$ intersects segment $\overline{zz_0}$ at distance less than ε from z_0 (see Figure 7).*

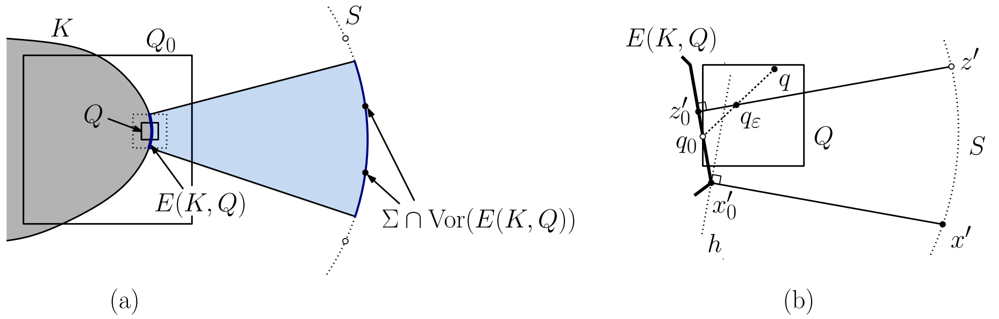
The following lemma is an extension of Dudley's results, which allows us to bound the complexity of an ε-approximation of K within a quadtree cell Q. Recall from section 2.2 that $E(K, Q)$ denotes the portion of ∂K that lies within distance $\sqrt{\varepsilon}$ of Q.

LEMMA 4.3. *Let K be a convex body, $Q \subseteq Q_0$ be a quadtree cell that intersects ∂K, and $0 < \varepsilon \leq 1/2$. Let Σ denote a set of $(\sqrt{\varepsilon}/4)$-dense points on the Dudley sphere S. Then $t(Q) \leq |\Sigma \cap \mathrm{Vor}(E(K, Q))|$ (see Figure 8(a)).*

Proof. We construct an approximating polytope P by the following local variant of Dudley's construction. For each point $x \in \Sigma \cap \mathrm{Vor}(E(K, Q))$, let x_0 be its nearest point on the boundary of K. (Note that $x_0 \in E(K, Q)$.) For each point x_0, take the supporting halfspace to K passing through x_0 that is orthogonal to the segment $\overline{xx_0}$. Let P be the (possibly unbounded) intersection of these halfspaces.

First, we show that $\Sigma \cap \mathrm{Vor}(E(K, Q))$ is nonempty. Consider any point z_0 on $\partial K \cap Q$. Let z denote any point of $S \cap \mathrm{Vor}(E(K, Q))$ whose closest point on ∂K is z_0. By definition of Σ, there is a point $x \in \Sigma$ whose distance from z is at most $\sqrt{\varepsilon}/4$. Letting x_0 denote x's closest point on ∂K, by Lemma 4.2(i), $\|z_0x_0\| \leq \sqrt{\varepsilon}/4 < \sqrt{\varepsilon}$. Thus, x_0 lies within $E(K, Q)$, which implies that $x \in \Sigma \cap \mathrm{Vor}(E(K, Q))$. It follows that P is bounded by at least one halfspace.

We now show that P is an (outer) ε-approximation of K within Q. Since P is defined by supporting hyperplanes, K is contained within P. Consider any $q \in Q$ that



(a) (b)

Fig. 8. *Lemma* 4.3. *(Not drawn to scale.)*

is at distance greater than ε from K. It suffices to show that $q \notin P$, that is, there exists a bounding hyperplane for P that separates q from K. Let q_0 denote the point of $K \cap Q$ that is closest to q (see Figure 8(b)). Note that q_0 is constrained to lie within Q, and hence this may not be the closest point to q on ∂K. By continuity, there must be a point on the segment $\overline{qq}_0$ that is at distance exactly ε from ∂K, which we denote by q_ε. Since Q is convex, this segment is contained in Q, and, hence, so is q_ε.

Let z'_0 be the point on ∂K that is closest to q_ε. (Note that z'_0 need not lie within Q.) Because Q_0 is centered at the origin, z'_0's distance from the origin is at most $\mathrm{diam}(Q_0)/2 + \|q_\varepsilon z'_0\| \le 1/2 + \varepsilon \le 1$. Let z' denote the point of intersection with the Dudley hypersphere S of the ray emanating from z'_0 and passing through q_ε. Let x' be a point of Σ that lies within distance $\sqrt{\varepsilon}/4$ of z', and let x'_0 be its closest point on ∂K. By Lemma 4.2(i) $\|x'_0 z'_0\| \le \sqrt{\varepsilon}/4$, and by (ii) the supporting hyperplane h at x'_0 orthogonal to the segment $\overline{x'x'_0}$ intersects segment $\overline{z'z'_0}$ at distance less than ε from z'_0. Thus, h separates q_ε from K, and therefore it separates q from K.

To complete the proof that $q \notin P$, it suffices to show that h is indeed included in our construction of P. By the triangle inequality and our assumption that $\varepsilon \le 1/2$, the distance from x'_0 to Q is at most

$$\|x'_0 z'_0\| + \|z'_0 q_\varepsilon\| \;\le\; \frac{\sqrt{\varepsilon}}{4} + \varepsilon \;\le\; \sqrt{\varepsilon}.$$

It follows that $x'_0 \in E(K,Q)$, and so h is included in the construction of P. By our hypothesis that the set $P(Q)$ constructed in step 3 of SplitReduce is the minimum-sized set of halfspaces needed to ε-approximate K within Q, we have $t(Q) = |P(Q)| \le |P| = |\Sigma \cap \mathrm{Vor}(E(K,Q))|$. (Note that this works even if Q is an "inside" cell that intersects K's boundary. In such a case $t(Q) = 1$ by definition, and as argued above, $\Sigma \cap \mathrm{Vor}(E(K,Q))$ is nonempty.) This completes the proof. □

Next, we prove a useful technical lemma, which bounds the total complexity of a set of leaves whose cells are of a given minimum size. Recalling the definition of Σ from the previous lemma, we may assume that $|\Sigma| = \Theta(1/\varepsilon^{(d-1)/2})$.

Lemma 4.4. *Let K be a convex body in $\mathbb{R}^d$, let $0 < \varepsilon \le 1/2$, and let L denote a set of disjoint quadtree cells contained within Q_0 such that each intersects ∂K and is of diameter $\Omega(\sqrt{\varepsilon})$. Then $\sum_{Q \in L} t(Q) = O(1/\varepsilon^{(d-1)/2})$.*

Proof. By applying Lemma 4.3 to each $Q \in L$ we have

$$\sum_{Q \in L} t(Q) \;\le\; \sum_{Q \in L} |\Sigma \cap \mathrm{Vor}(E(K,Q))|.$$

Since $|\Sigma| = O(1/\varepsilon^{(d-1)/2})$, to complete the proof it suffices to show that each $x \in \Sigma$ lies within $\mathrm{Vor}(E(K,Q))$ for at most constant number of $Q \in L$. To see this, let x_0 be the point on ∂K that is closest to x. Since each cell $Q \in L$ has size at least $\Omega(\sqrt{\varepsilon})$, by disjointness and a packing argument it follows that at most a constant number (depending on dimension) of such cells can lie within distance $\sqrt{\varepsilon}$ of x_0, which establishes the claim. ∎

Combining the above results, we obtain the main result of this section.

LEMMA 4.5. *Let K be a convex body in $\mathbb{R}^d$ and $0 < \varepsilon \leq 1/2$. The output of SplitReduce(K, Q_0) for $t \geq 1/\varepsilon^{(d-1)/4}$ has total space $O(1/\varepsilon^{(d-1)/2})$.*

Proof. Let c_2 be the constant of Lemma 4.1, and define $c_1 = (1/c_2)^{2/(d-1)}$. We may assume that $\varepsilon \leq c_1^2$, for otherwise $\varepsilon = \Omega(1)$ and clearly SplitReduce will not generate more than a constant number of cells.

Let T denote the quadtree produced by the algorithm, and let L denote the set of leaf cells of T that intersect the boundary of K. Recall from Lemma 3.1 that the data structure's total space is asymptotically bounded by the sum of $t(Q)$ for all $Q \in L$. Thus, it suffices to prove that

$$\sum_{Q \in L} t(Q) \;=\; O(1/\varepsilon^{(d-1)/2}).$$

Toward this end, we first prove a lower bound on the size of any leaf cell Q. We assert that the cell Q associated with any internal node has diameter at least $\delta = c_1\sqrt{\varepsilon}$. It will then follow that each leaf cell has diameter at least $\delta/2$. Suppose to the contrary that $\mathrm{diam}(Q) < \delta$. Recall the standardization transformation from section 2.2, which maps Q to Q_0 and scales ε to at least $\varepsilon/\delta = \sqrt{\varepsilon}/c_1$. Let us denote this value by ε'. Since $\varepsilon \leq c_1^2$, we have $\varepsilon' \leq 1$. By applying Lemma 4.1 to the transformed body (with ε' playing the role of ε), it follows that the polytope $K \cap Q$ can be ε-approximated by a polytope P defined by the intersection of at most

$$\frac{c_2}{(\varepsilon')^{(d-1)/2}} \;=\; c_2\left(\frac{c_1}{\sqrt{\varepsilon}}\right)^{(d-1)/2} \;\leq\; \frac{1}{\varepsilon^{(d-1)/4}}$$

halfspaces. Since $K \cap Q \subseteq P$, it is easy to see that P is an ε-approximation of K within Q. Since $t \geq 1/\varepsilon^{(d-1)/4}$, the termination condition of our algorithm implies that such a cell is not further subdivided, contradicting our hypothesis that this is an internal node. Therefore, the cells of L satisfy the conditions of Lemma 4.4. The desired bound follows by applying this lemma. ∎

It is useful to contrast this with the Dudley-based approach described in section 2.4. For $t = 1/\varepsilon^{(d-1)/4}$, we obtain the same $O(1/\varepsilon^{(d-1)/2})$ space in each case, but the exponent in the query time of SplitReduce is only half that of the Dudley-based approach. Later, in Lemma 6.3, we will present a more refined analysis showing that it is possible to reduce this further, achieving a query time of only $\widetilde{O}(1/\varepsilon^{(d-1)/8})$.

It will be useful in later sections to generalize the above lemma to quadtree cells of arbitrary size. By a direct application of standardization, we obtain the following.

LEMMA 4.6. *Let K be a convex body in $\mathbb{R}^d$, let Q be a quadtree cell contained within Q_0, and let $0 < \varepsilon \leq \mathrm{diam}(Q)/2$. The output of the call SplitReduce(K, Q) for $t \geq (\mathrm{diam}(Q)/\varepsilon)^{(d-1)/4}$ has total space $O((\mathrm{diam}(Q)/\varepsilon)^{(d-1)/2})$.*

5. Dual caps and approximation.

The bounds proved in the previous section apply to query times $t \geq 1/\varepsilon^{(d-1)/4}$. In section 6 we will show how to obtain good

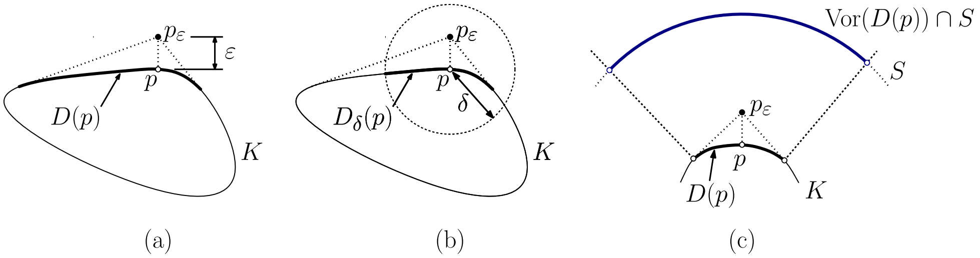


FIG. 9. (a) *Dual caps,* (b) *restricted dual caps, and* (c) *the Voronoi patch of a dual cap.*

space bounds for smaller query times. This will involve analyzing the local geometry about small boundary patches of the convex body. In this section, we introduce the principal geometric underpinnings that will be needed for this more refined analysis. In particular, we discuss the concepts of dual caps and restricted dual caps and their role in polytope approximation.

Although we do not assume that K is smooth, it will simplify the presentation to imagine that each boundary point has a unique supporting hyperplane and a unique normal vector. To achieve this, we employ an *augmented representation* of the boundary points of K. In particular, each boundary point $p \in \partial K$ will be expressed as a pair (p, h), where h is a supporting hyperplane at p. We will often refer to h as $h(p)$. When h is clear from context or unimportant, we avoid explicit reference to it.

We first observe that computing an outer ε-approximation of a convex body K by halfspaces can be reduced to a hitting-set problem. Consider any point p_ε that is external to K at distance ε from its boundary, and let (p, h) denote the augmented boundary point consisting of the closest point $p \in \partial K$ to p_ε and the supporting hyperplane through p that is orthogonal to the segment pp_ε (see Figure 9(a)). We define the *ε-dual cap* of p, denoted $D(p)$, to be the set of augmented boundary points (q, h') such that the supporting hyperplane h' through q intersects the closed line segment pp_ε. (Equivalently, these are the points of ∂K that are visible to p_ε.)

Any outer ε-approximation of K by halfspaces must contain at least one halfspace that separates p from p_ε, and this can be achieved by including h' for any pair (q, h') within $D(p)$. A set of augmented points $\Sigma \subseteq \partial K$ is said to be an *ε-hitting set* for K if for every $p \in \partial K$, $\Sigma \cap D(p) \neq \emptyset$. It follows directly that the intersection of the supporting halfspaces for any ε-hitting set is an outer ε-approximation of K. This observation will be formalized within our quadtree-based context in our next lemma. Before stating the lemma, we need to introduce one additional concept. In order to approximate K within a given quadtree cell Q, we are interested only in the geometry of K's boundary that lies close to Q. For this reason, it will be desirable to limit the diameter of dual caps. Given $\delta > 0$, let $B_\delta(p)$ denote the closed Euclidean ball of radius δ centered at p. Define the *δ-restricted dual cap*, denoted $D_\delta(p)$, to be the intersection of $D(p)$ with $B_\delta(p)$ (see Figure 9(b)).

LEMMA 5.1. *Let K be a convex body, let $Q \subseteq Q_0$ be a quadtree cell that intersects ∂K, and $0 < \varepsilon \leq 1/2$. Let Σ be any set of augmented points on $E(K, Q)$ that hits the set of all $\sqrt{\varepsilon}$-restricted ε-dual caps whose defining point is in $E(K, Q)$ (see Figure* 10(a)*). Then there is a polytope P defined as the intersection of $|\Sigma|$ halfspaces that ε-approximates K within Q.*

Proof. Let P be the polytope defined by the intersection of the supporting halfspaces associated with each augmented point of Σ (see Figure 10(b)). Clearly,

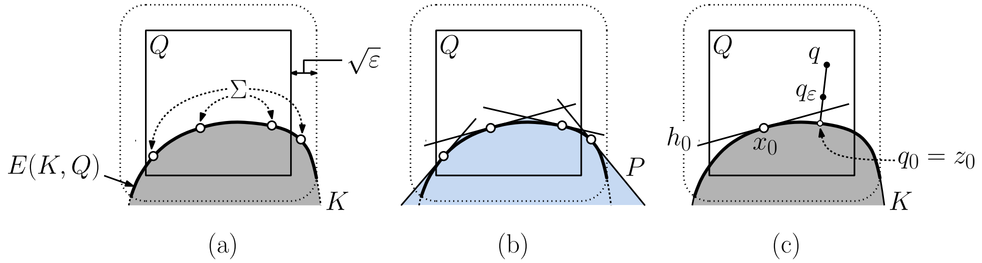


FIG. 10. *Lemma* 5.1.

$K \subseteq P$. Consider any point $q \in Q$ that is at distance greater than ε from $K \cap Q$. It suffices to show that $q \notin P$, that is, there exists a bounding hyperplane for P that separates q from K.

We apply a similar argument to the one that we used in the proof of Lemma 4.3. Consider any $q \in Q$ that is at distance greater than ε from K (see Figure 10(c)). It suffices to show that there exists a bounding hyperplane for P that separates q from K. Let q_0 denote the point of $K \cap Q$ that is closest to q. By continuity, there must be a point on the segment $\overline{qq_0}$ that is at distance exactly ε from ∂K, which we denote by q_ε. Since Q is convex, this segment must be contained in Q, and, hence, so is q_ε.

Let z_0 be the point on ∂K that is closest to q_ε. (In our figure $z_0 = q_0$, but generally z_0 need not lie within Q.) Since $\varepsilon \leq 1$, we have $\|q_\varepsilon z_0\| = \varepsilon \leq \sqrt{\varepsilon}$. It follows that $z_0 \in E(K,Q)$. Therefore, there exists an augmented point $(x_0, h_0) \in \Sigma$ that hits the $\sqrt{\varepsilon}$-restricted ε-dual cap defined by z_0 (whose apex is at q_ε). The supporting hyperplane h_0 separates q_ε (and therefore q) from K, as desired. □

Our analysis of the space bounds of SplitReduce is based on the combined sizes of the ε-hitting sets for K within each quadtree cell Q. Dudley's construction can be viewed as one method of computing ε-hitting sets. Unfortunately, Dudley's construction does not lead to the best bounds because it tends to oversample in regions of very low or very high curvature. Our analysis will be based on a more refined, area-based approach to bounding the sizes of hitting sets. The key geometric observation is that the product of the areas of any ε-dual cap and its associated Voronoi patch on the Dudley sphere S must be large. Intuitively, if the surface area of an ε-dual cap is small, then the total curvature of the patch must be high, and so the associated Voronoi patch must have relatively large area (see Figure 9(c)). More precisely, we show that (under certain conditions) the product of the areas of an ε-dual cap and its Voronoi patch is $\Omega(\varepsilon^{d-1})$. This result is stated formally in Lemma 5.2 below. Given a $(d-1)$-dimensional manifold, let area(Y) denote its $(d-1)$-dimensional Hausdorff measure. Given a convex body X in $\mathbb{R}^d$, we use area(X) as a shorthand for area(∂X).

LEMMA 5.2 (area-product bound). *Let K be a convex body in $\mathbb{R}^d$, and let $0 < \varepsilon \leq 1/8$. Consider a pair $(p, h(p))$, where $p \in \partial K$ and $h(p)$ is a supporting hyperplane passing through p. Let D denote the $\sqrt{\varepsilon}$-restricted ε-dual cap whose defining point is p. If K is fat and of diameter at least 2ε, there exists a constant c_a (depending only on d) such that if p lies within a unit ball centered at the origin, then* area$(D) \cdot$ area$(\mathrm{Vor}(D) \cap S) > c_a \cdot \varepsilon^{d-1}$.

The proof of the lemma is quite technical and will be deferred to section 10. The geometric basis of the proof involves the Mahler volume, which was introduced in section 2.3. The bound stated in the lemma holds if K is γ-fat for any γ in the interval $(0,1]$ under the assumption that γ does not depend on ε. In particular, the proof will reveal that $c_a = \Omega(\gamma^{d-1})$.

We will exploit this observation to demonstrate the existence of smaller ε-hitting sets than those given by Dudley's construction. We will hit the restricted ε-dual caps that have large surface area by sampling points randomly on the boundary of K, and we will hit those with small surface area by sampling points randomly on the Dudley hypersphere and then selecting their nearest neighbors on ∂K. In order to prove that such a random sampling strategy works to stab all the dual caps, we need to establish bounds on the VC-dimension of an appropriate range space based on dual caps. This is not surprising given that dual caps and restricted dual caps are defined by a constant number of parameters. The result is stated in the following lemma. The proof involves a straightforward application of basic geometric principles. (Details can be found in [8].)

Lemma 5.3. *Let K be a convex body in $\mathbb{R}^d$ that lies within Q_0, and let ε and δ be positive real parameters. The following range spaces (X_i, R_i) have constant VC-dimension (where the constant depends only on d):*

1. *$X_1 = \partial K$ and R_1 is the set of ε-dual caps.*
2. *$X_2 = S$ and R_2 is the set of Voronoi patches of the ε-dual caps.*
3. *$X_3 = \partial K$ and R_3 is the set of δ-restricted ε-dual caps.*
4. *$X_4 = S$ and R_4 is the set of Voronoi patches of the δ-restricted ε-dual caps.*

In the next section we will exploit this result to establish the existence of small ε-nets for these range spaces. Note that the range spaces defined in this lemma are defined over ∂K, a domain of infinite cardinality. However, for our purposes, it suffices to consider dual caps and restricted dual caps whose defining points are drawn from any sufficiently dense set of points on ∂K (depending on ε), and therefore the domains of the range spaces can be treated as finite sets.

6. Final upper bound.

In this section, we use the tools developed in sections 4 and 5 to obtain better upper bounds for approximate polytope membership. In particular, we present a proof of Theorem 1.2. We will first show how to apply the area-based techniques described in the previous section to improve the simple upper bound from Lemma 4.5 at the low-space end of the trade-off spectrum. (This will be presented in Lemma 6.3.) We will then apply this improvement repeatedly in an inductive manner to establish trade-offs throughout the spectrum. For technical reasons, many of the lemmas of this section assume constant upper bounds on the value of ε. There is no loss of generality in doing so, since it is easy to show that if ε is bounded below by any fixed constant, the asymptotic space and query times of SplitReduce are both $O(1)$.

Throughout this section, recall that $E_\delta(K,Q)$ is the portion of ∂K that is within distance δ of Q, and $E(K,Q) = E_{\sqrt{\varepsilon}}(K,Q)$. Also, define $E^+(K,Q) = E_{2\sqrt{\varepsilon}}(K,Q)$. We will assume that $\mathrm{diam}(K) \geq 2\varepsilon$, for otherwise it is trivial to compute an ε-approximation of constant size. Our first result establishes an area-based bound on the number of halfspaces needed to approximate K within a quadtree cell Q.

Lemma 6.1. *Let K be a fat convex body in $\mathbb{R}^d$, let $0 < \varepsilon \leq 1/8$, and let $Q \subseteq Q_0$ be a quadtree cell that intersects ∂K. Letting c_a denote the constant of Lemma 5.2, define*

$$r \;=\; \left(\frac{\mathrm{area}(E^+(K,Q)) \cdot \mathrm{area}\left(\mathrm{Vor}(E^+(K,Q)) \cap S\right)}{c_a \cdot \varepsilon^{d-1}}\right)^{1/2}.$$

There is a polytope P defined as the intersection of $O(r \log r)$ halfspaces that is an ε-approximation of K within Q.

Proof. Letting $A_K = \text{area}(E^+(K,Q))$ and $A_S = \text{area}\,(\text{Vor}(E^+(K,Q)) \cap S)$, we can express the value of r more succinctly as $(A_K A_S / c_a\, \varepsilon^{d-1})^{1/2}$. First, we assert that $r = \Omega(1)$. To see this, we consider two cases. First, if K lies entirely within distance $2\sqrt{\varepsilon}$ of Q, then $\text{Vor}(E^+(K,Q)) \cap S = S$, which implies that $A_S = \Omega(1)$. Since K is fat and by our assumption that $\text{diam}(K) \geq 2\varepsilon$, it follows that $A_K = \Omega(\varepsilon^{d-1})$. Therefore, $r = \Omega(1)$. On the other hand, if some part of K lies at distance greater than $2\sqrt{\varepsilon}$ from Q, $E^+(K,Q)$ is a boundary patch of K of diameter $\Omega(\sqrt{\varepsilon})$. Since both K and Q are fat, it follows that $A_K = \Omega(\varepsilon^{(d-1)/2})$. By convexity, as we go from a boundary patch on K to its Voronoi cell on S, distances cannot decrease. Therefore $A_S \geq A_K$, and again we have $r = \Omega(1)$. Through a minor adjustment to constant factor c_a in r's definition, we may assume that $\log r \geq 1$.

By Lemma 5.1, in order to show the existence of an ε-approximating polytope P for K within Q, it suffices to show that it is possible to hit all $\sqrt{\varepsilon}$-restricted ε-dual caps whose defining point lies in $E(K,Q)$ (not to be confused with $E^+(K,Q)$) using $O(r \log r)$ points. To do this, we distinguish between two types of such restricted dual caps. A restricted dual cap D is of *type* 1 if $\text{area}(D) \geq (c_a\, \varepsilon^{d-1} A_K / A_S)^{1/2}$, and otherwise it is of *type* 2.

By assertions 3 and 4 of Lemma 5.3, we know that $\sqrt{\varepsilon}$-restricted ε-dual caps and their Voronoi patches both have constant VC-dimension. The VC-dimension is no larger if we restrict the domain of the range space. Therefore, by standard machinery (see, e.g., [4]) we can build a $(1/r)$-net for any restriction of these range spaces of size $O(r \log r)$ each by random sampling.

For type-1 dual caps, consider the restriction $(E^+(K,Q), R_3)$ of the range space given in Lemma 5.3.3. Let Σ_1 denote a $(1/r)$-net. Consider any type-1 dual cap D. Since D's defining point lies within $E(K,Q)$ and it is $\sqrt{\varepsilon}$-restricted, it lies entirely within $E^+(K,Q)$. Thus, we have

$$\frac{\text{area}(D \cap E^+(K,Q))}{\text{area}(E^+(K,Q))} = \frac{\text{area}(D)}{\text{area}(E^+(K,Q))} \;\geq\; \frac{\left(c_a \cdot \varepsilon^{d-1} A_K / A_S\right)^{1/2}}{A_K}$$
$$= \left(\frac{c_a \cdot \varepsilon^{d-1}}{A_K A_S}\right)^{1/2} \;=\; \frac{1}{r}.$$

Therefore D contains at least one point of Σ_1. It follows that Σ_1 hits all type-1 dual caps.

For type-2 dual caps, let us consider the restriction $(\text{Vor}(E^+(K,Q)) \cap S, R_4)$ of the range space of Lemma 5.3.4. Let Σ_2 denote a $(1/r)$-net. Because $\varepsilon \leq 1/8$ and $Q \subseteq Q_0$, $E(K,Q)$ lies within a ball centered at the origin of radius $\text{diam}(Q_0)/2 + \sqrt{\varepsilon} \leq 1$. Given any type-2 dual cap D whose defining (augmented) point lies in $E(K,Q)$, we may apply Lemma 5.2 to obtain

$$\text{area}(\text{Vor}(D) \cap S) \;\geq\; \frac{c_a \cdot \varepsilon^{d-1}}{\text{area}(D)} \;\geq\; \frac{c_a \cdot \varepsilon^{d-1}}{\left(c_a \cdot \varepsilon^{d-1} A_K / A_S\right)^{1/2}} \;=\; \left(\frac{c_a \cdot \varepsilon^{d-1} A_S}{A_K}\right)^{1/2}.$$

As before, since D's defining point lies within $E(K,Q)$, $D \subseteq E^+(K,Q)$. From this we have

$$\begin{aligned}\frac{\operatorname{area}(\operatorname{Vor}(D \cap E^+(K,Q)) \cap S)}{\operatorname{area}(\operatorname{Vor}(E^+(K,Q)) \cap S)} &= \frac{\operatorname{area}(\operatorname{Vor}(D) \cap S)}{\operatorname{area}(\operatorname{Vor}(E^+(K,Q)) \cap S)}\\ &\geq \frac{\left(c_a \cdot \varepsilon^{d-1} A_S/A_K\right)^{1/2}}{A_S} \;=\; \left(\frac{c_a \cdot \varepsilon^{d-1}}{A_K A_S}\right)^{1/2}\\ &= \frac{1}{r}.\end{aligned}$$

Therefore $\operatorname{Vor}(D) \cap S$ contains at least one point of Σ_2, implying that Σ_2 hits the Voronoi patches of all type-2 dual caps. For each point of Σ_2, we select its nearest neighbor on ∂K, obtaining a set $\Sigma_2' \subset E^+(K,Q)$. It follows directly that the set Σ_2' hits all type-2 dual caps. Therefore, the union $\Sigma_1 \cup \Sigma_2'$ forms the desired set of size $O(r \log r)$ that hits all $\sqrt{\varepsilon}$-restricted ε-dual caps whose defining point lies within $E(K,Q)$. ∎

In order to establish our storage bounds, we analyze the behavior of the algorithm at a particular level of the decomposition. Given the query-time parameter t, recall that we stop the subdivision process in SplitReduce(K,Q) if the number of hyperplanes needed to approximate K within Q falls below t. Also recall that $t(Q)$ denotes the number of approximating halfspaces associated with Q. Let us consider the state of the subdivision process when the cell sizes reach roughly $\sqrt{\varepsilon}$. Cells that have stopped subdividing by this point are "good," since we can bound the total space requirements for all such cells by appealing to Lemma 4.4. For the remaining "bad" cells, we will bound their space requirements on a cell-by-cell basis by using the simple upper bound from Lemma 4.6. For our approach to work well, it is crucial to obtain a good bound on the number of such bad cells. We exploit the area bound of Lemma 6.1 for this purpose. Whenever SplitReduce subdivides a cell of size $O(\sqrt{\varepsilon})$, we can infer that more than t hyperplanes are required to approximate K within this cell. Since the portion of ∂K lying within this cell is small, the area of its Voronoi patch on the Dudley sphere must be large. A packing argument applied on the Dudley sphere will be used to limit the number of these bad cells.

In order to formalize the notion of good and bad cells, let T denote the quadtree produced by SplitReduce(K, Q_0), and let T' denote the subtree of T induced by cells of diameter at least $\sqrt{\varepsilon}/2$. For the remainder of this section, let L_1 denote the (good) leaf cells of T' that are not subdivided further by the algorithm, and let L_2 be the remaining (bad) leaf cells of T'. The cells of L_1 and L_2 are all of diameter $\Omega(\sqrt{\varepsilon})$. Each cell in L_1 can be approximated using at most t halfspaces, and those in L_2 require more. In our next lemma, we bound the total number of approximating halfspaces over all the good leaf cells and the total number of bad leaf cells.

Lemma 6.2. *Let K be a fat convex body in $\mathbb{R}^d$, and let $0 < \varepsilon \leq 1/8$. Let T denote the quadtree produced by SplitReduce(K, Q_0) for $t \geq 1$, and let L_1 and L_2 be as defined above. Then*

(i) $\sum_{Q \in L_1} t(Q) = O(1/\varepsilon^{(d-1)/2})$,

(ii) $|L_2| = O((1 + \log t)/t)^2 (1/\varepsilon)^{(d-1)/2})$.

Proof. Because the cells of L_1 are disjoint and each is of diameter $\Omega(\sqrt{\varepsilon})$, assertion (i) follows as a direct consequence of Lemma 4.4. Thus, it remains to prove assertion (ii). Let Q be any cell of L_2. Since any child of a cell of L_2 is of diameter smaller than $\sqrt{\varepsilon}/2$ and Q's diameter is twice this, we have $\sqrt{\varepsilon}/2 \leq \operatorname{diam}(Q) < \sqrt{\varepsilon}$. Recall that $E^+(K,Q) = E_{2\sqrt{\varepsilon}}(K,Q)$. Also, let $A_K(Q)$ and $A_S(Q)$ denote the values of A_K and A_S, respectively, from the proof of Lemma 6.1, when applied to Q.

Because $\mathrm{diam}(Q) \leq \sqrt{\varepsilon}$ and $E^{+}(K,Q)$ involves a boundary patch of K that intersects Q and includes an additional expansion by distance $2\sqrt{\varepsilon}$, it follows that this boundary patch has diameter $O(\sqrt{\varepsilon})$. Therefore, $A_K(Q) = O(\varepsilon^{(d-1)/2})$. By applying Lemma 6.1 (and recalling the constant c_a from Lemma 5.2), we have $t(Q) = O(r \log r)$, where

$$\begin{aligned} r &= \left(\frac{\mathrm{area}(E^{+}(K,Q)) \cdot \mathrm{area}\,(\mathrm{Vor}(E^{+}(K,Q)) \cap S)}{c_a \cdot \varepsilon^{d-1}}\right)^{1/2} \\ &= \left(\frac{A_K(Q) A_S(Q)}{c_a \cdot \varepsilon^{d-1}}\right)^{1/2} \;=\; O\left(\sqrt{\frac{A_S(Q)}{\varepsilon^{(d-1)/2}}}\right). \end{aligned}$$

In Lemma 6.1 we showed that (after a suitable adjustment to c_a) we have $\log r \geq 1$. Since Q is subdivided further, we know that $t(Q) > t$, which implies that $t = O(r \log r)$. Because $t \geq 1$, by simple manipulations we have $t/(1+\log t) = O(r)$. By combining this with the upper bound on r from above, we obtain $A_S(Q) = \Omega((t/(1+\log t))^2 \varepsilon^{(d-1)/2})$, which yields the lower bound

$$\sum_{Q \in L_2} A_S(Q) \;=\; |L_2| \cdot \Omega\left(\left(\frac{t}{1+\log t}\right)^2 \varepsilon^{\frac{d-1}{2}}\right).$$

As shown in the proof of Lemma 4.4, given any set of disjoint quadtree cells of diameter $\Omega(\sqrt{\varepsilon})$ a point of S can be in $\mathrm{Vor}(E^{+}(K,Q))$ for at most a constant number of these cells. Since the quadtree cells of L_2 satisfy these conditions,

$$\sum_{Q \in L_2} A_S(Q) \;=\; \sum_{Q \in L_2} \mathrm{area}(\mathrm{Vor}(E^{+}(K,Q)) \cap S) \;=\; O(\mathrm{area}(S)).$$

Combining this with our lower bound, we have

$$|L_2| \;=\; O\left(\mathrm{area}(S) \cdot \left(\frac{1+\log t}{t}\right)^2 \cdot \left(\frac{1}{\varepsilon}\right)^{\frac{d-1}{2}}\right).$$

Since S is a hypersphere of constant radius, its area is bounded, and assertion (ii) follows immediately. ∎

Recall that we showed in Lemma 4.5 that it is possible to answer approximate membership queries in $1/\varepsilon^{(d-1)/4}$ time using space $O(1/\varepsilon^{(d-1)/2})$. By using the above lemma, we show next that we can improve this to achieving query time roughly $O(1/\varepsilon^{(d-1)/8})$ for the same space.

LEMMA 6.3. *Let K be a fat convex body in $\mathbb{R}^d$, and let $0 < \varepsilon \leq 1/16$. For $t \geq (\lg \frac{1}{\varepsilon})/\varepsilon^{(d-1)/8}$, the output of SplitReduce(K, Q_0) has total space $O(1/\varepsilon^{(d-1)/2})$.*

Proof. Let T denote the quadtree produced by the algorithm. By Lemma 3.1, the data structure's total space is dominated by the space needed to store the hyperplanes in the leaf cells. Thus, it suffices to show that the sum of $t(Q)$ over all leaf cells Q of T is $O(1/\varepsilon^{(d-1)/2})$. Let T', L_1, and L_2 be as defined just prior to Lemma 6.2. By Lemma 6.2(i), the total contribution of $t(Q)$ for all cells in L_1 is $O(1/\varepsilon^{(d-1)/2})$. So, it suffices to bound the contribution due to L_2.

Let Q be any cell of L_2. Recall from the proof of Lemma 6.2 that $\sqrt{\varepsilon}/2 \leq \mathrm{diam}(Q) \leq \sqrt{\varepsilon}$. Since $t \geq 1/\varepsilon^{(d-1)/8}$, it follows that $t \geq (\mathrm{diam}(Q)/\varepsilon)^{(d-1)/4}$. Because $\varepsilon \leq 1/16$, we have $\varepsilon \leq \sqrt{\varepsilon}/4 \leq \mathrm{diam}(Q)/2$. By Lemma 4.6, the output of SplitReduce(K, Q) has total space at most

$$O\left(\left(\frac{\mathrm{diam}(Q)}{\varepsilon}\right)^{\frac{d-1}{2}}\right) ~=~ O\left(\left(\frac{1}{\varepsilon}\right)^{\frac{d-1}{4}}\right).$$

By Lemma 6.2(ii), $|L_2| = O(((1+\log t)/t)^2(1/\varepsilon)^{(d-1)/2})$. Since $t \geq (\lg \frac{1}{\varepsilon})/\varepsilon^{(d-1)/8}$, we have $|L_2| = O(1/\varepsilon^{(d-1)/4})$. Summing up the space contributions of all $Q \in L_2$, the total space for these cells is

$$|L_2| \cdot O\big(1/\varepsilon^{(d-1)/4}\big) ~=~ O\big(1/\big(\varepsilon^{(d-1)/4} \cdot \varepsilon^{(d-1)/4}\big)\big) ~=~ O\big(1/\varepsilon^{(d-1)/2}\big),$$

as desired. ☐

In order to extend the space-time trade-off to other query times, we will apply the previous result as the basis case in an induction argument. The induction will be controlled by a parameter α, which we assume to be a constant. The proof is rather technical, but it involves a straightforward application of the earlier results of this section.

LEMMA 6.4. *Let K be a fat convex body in $\mathbb{R}^d$, and let $0 < \varepsilon \leq 1/16$. Let $\alpha \geq 4$ be a real-valued constant. For $t \geq (\lg \frac{1}{\varepsilon})/\varepsilon^{(d-1)/\alpha}$, the output of SplitReduce(K, Q_0) has total space*

$$O\left(1/\varepsilon^{(d-1)\left(1-2\left(\frac{\lfloor \lg \alpha \rfloor - 2}{\alpha} + \frac{1}{2^{\lfloor \lg \alpha \rfloor}}\right)\right)}\right).$$

Proof. Define $k = \lfloor \lg \alpha \rfloor$, which implies that $k \geq 2$, and $2^k \leq \alpha < 2^{k+1}$. Expressed as a function of k, the desired space bound can be expressed as

$$c_k \cdot \left(1/\varepsilon^{(d-1)\left(1-2\left(\frac{k-2}{\alpha}+\frac{1}{2^k}\right)\right)}\right) \tag{1}$$

for a constant c_k (depending on k but not on ε).

We begin exactly as in the proof of the previous lemma. Let T denote the quadtree produced by the algorithm, and by Lemma 3.1, it suffices to bound the sum of $t(Q)$ over all leaf cells of T. Given T', L_1, and L_2 defined prior to Lemma 6.2, the space contribution due to the cells of L_1 is $O(1/\varepsilon^{(d-1)/2})$. To see that this satisfies our space bound, observe that since $k \geq 2$ and $\alpha \geq 2^k$, we have

$$\frac{1}{4} ~\geq~ \frac{k-1}{2^k} ~=~ \frac{k-2}{2^k} + \frac{1}{2^k} ~\geq~ \frac{k-2}{\alpha} + \frac{1}{2^k}.$$

Therefore, the total contribution of $t(Q)$ for all cells in L_1 is

$$O\big(1/\varepsilon^{(d-1)/2}\big) ~=~ O\left(1/\varepsilon^{(d-1)\left(1-2\left(\frac{1}{4}\right)\right)}\right) ~\leq~ O\left(1/\varepsilon^{(d-1)\left(1-2\left(\frac{k-2}{\alpha}+\frac{1}{2^k}\right)\right)}\right), \tag{2}$$

which matches the desired bound given in (1).

It remains to bound the contribution to the space of the cells of L_2. We do this by induction on k. For the basis case $k = 2$, we have $4 \leq \alpha < 8$. Therefore $t > (\lg \frac{1}{\varepsilon})/\varepsilon^{(d-1)/8}$. By applying Lemma 6.3, the total space of the data structure (which includes the contribution of L_2) is $O(1/\varepsilon^{(d-1)/2})$. It follows from (2) (for the case $k = 2$) that this satisfies our storage bound.

For the induction step, we assume that the lemma holds for $k-1$ (that is, $2^{k-1} \leq \alpha/2 < 2^k$), and our objective is to prove it for k. It will be convenient to express the induction hypothesis in a form that holds for an arbitrary quadtree cell $Q \subseteq Q_0$.

By applying standardization to Q (thus mapping Q to Q_0 and scaling ε to $\varepsilon/\mathrm{diam}(Q)$), the induction hypothesis states that for

$$0 \;<\; \varepsilon \;\leq\; \frac{\mathrm{diam}(Q)}{16} \quad \text{and} \quad t \;\geq\; \left(\lg \frac{\mathrm{diam}(Q)}{\varepsilon}\right) \cdot \left(\frac{\mathrm{diam}(Q)}{\varepsilon}\right)^{\frac{d-1}{\alpha/2}}, \tag{3}$$

there is a constant c_{k-1} such that the output of SplitReduce(K, Q) has total space at most

$$c_{k-1} \cdot (\mathrm{diam}(Q)/\varepsilon)^{(d-1)\left(1-2\left(\frac{k-3}{\alpha/2}+\frac{1}{2^{k-1}}\right)\right)}. \tag{4}$$

Let Q be any cell of L_2. In the proof of Lemma 6.2 we showed that $\sqrt{\varepsilon}/2 \leq \mathrm{diam}(Q) < \sqrt{\varepsilon}$. By the bound on t from the statement of this lemma, we have

$$t \;\geq\; \left(\lg \frac{1}{\varepsilon}\right)\left(\frac{1}{\varepsilon}\right)^{\frac{d-1}{\alpha}} \;\geq\; \left(\lg \frac{1}{\sqrt{\varepsilon}}\right)\left(\frac{1}{\sqrt{\varepsilon}}\right)^{\frac{2(d-1)}{\alpha}} \;\geq\; \left(\lg \frac{\mathrm{diam}(Q)}{\varepsilon}\right)\left(\frac{\mathrm{diam}(Q)}{\varepsilon}\right)^{\frac{d-1}{\alpha/2}},$$

implying that t satisfies (3). If ε is at most $\mathrm{diam}(Q)/16$, we may apply the induction hypothesis, yielding the space bound given in (4). Since $\mathrm{diam}(Q) < \sqrt{\varepsilon}$, this can be simplified to

$$c_{k-1} \cdot (1/\sqrt{\varepsilon})^{(d-1)\left(1-2\left(\frac{k-3}{\alpha/2}+\frac{1}{2^{k-1}}\right)\right)} \;=\; c_{k-1} \cdot 1/\varepsilon^{(d-1)\left(\frac{1}{2}-\frac{2(k-3)}{\alpha}-\frac{1}{2^{k-1}}\right)}. \tag{5}$$

By combining Lemma 6.2(ii) with the lower bound on t given in the statement of this lemma, the number of cells in L_2 satisfies

$$|L_2| \;=\; O\!\left(\left(\frac{\lg t}{t}\right)^2 \left(\frac{1}{\varepsilon}\right)^{\frac{d-1}{2}}\right) \;=\; O\!\left(\varepsilon^{\frac{2(d-1)}{\alpha}} \left(\frac{1}{\varepsilon}\right)^{\frac{d-1}{2}}\right) \;=\; O\!\left(\left(\frac{1}{\varepsilon}\right)^{(d-1)\left(\frac{1}{2}-\frac{2}{\alpha}\right)}\right). \tag{6}$$

The total contribution to the space by the cells of L_2 is the product of the space requirements for each cell of L_2, given in (5), and the number of such cells, given in (6). There exists a constant c_k (depending on k but not on ε) such that the total space is at most

$$c_k \cdot \left(1/\varepsilon^{(d-1)\left(\left(\frac{1}{2}-\frac{2(k-3)}{\alpha}-\frac{1}{2^{k-1}}\right)+\left(\frac{1}{2}-\frac{2}{\alpha}\right)\right)}\right) \;=\; c_k \cdot \left(1/\varepsilon^{(d-1)\left(1-2\left(\frac{k-2}{\alpha}+\frac{1}{2^k}\right)\right)}\right).$$

On the other hand, if ε exceeds $\mathrm{diam}(Q)/16$, then since $\mathrm{diam}(Q) \geq \sqrt{\varepsilon}/2$ it follows that ε is $\Omega(1)$, and we can adjust to c_k to satisfy this bound. In either case, we achieve the bound in (1). ◻

Observe that the exponent in the space bound in the preceding lemma is a piecewise linear function in $1/\alpha$, whose breakpoints coincide with powers of two. It is easily verified that the exponent is a continuous function of α. (In particular, observe that $\lim_{\delta\to 0} f(2^{k-\delta}) = f(2^k)$, where $f(\alpha) = 1/2^{\lfloor \lg \alpha \rfloor} + (\lfloor \lg \alpha \rfloor - 2)/\alpha$.)

We can now present the proof of Theorem 1.2. Recall that K is a convex polytope in $\mathbb{R}^d$. By Lemma 2.1, we can precondition K so that it is $(1/d)$-fat and is contained within Q_0, thus allowing us to approximate K absolutely. Also, if $1/16 < \varepsilon \leq 1$, we set $\varepsilon = 1/16$. (Both of these changes result in a constant factor decrease to ε, which will not affect the asymptotic bounds.) We then set $t = (\lg \frac{1}{\varepsilon})/\varepsilon^{(d-1)/\alpha}$ and invoke SplitReduce(K, Q_0). Let T denote the resulting data structure. Given the preconditioning of K and the alteration of ε, we may apply Lemma 6.4 to show that the total space for T is

$$O\left(1/\varepsilon^{(d-1)\left(1-2\left(\frac{\lfloor \lg \alpha \rfloor - 2}{\alpha}+\frac{1}{2^{\lfloor \lg \alpha \rfloor}}\right)\right)}\right).$$

Using the fact that $1/2^{\lfloor \lg \alpha \rfloor} \geq 1/\alpha$, this is

$$O\left(1/\varepsilon^{(d-1)\left(1-2\left(\frac{\lfloor \lg \alpha \rfloor - 2}{\alpha}+\frac{1}{\alpha}\right)\right)}\right) \;=\; O\left(1/\varepsilon^{(d-1)\left(1-\frac{2\lfloor \lg \alpha \rfloor - 2}{\alpha}\right)}\right),$$

which matches the space bound of Theorem 1.2.

Recall that a query is answered by locating the leaf node of T that contains the query point, followed by an inspection of the (at most) t halfspaces stored in this leaf node. By our remarks following the presentation of SplitReduce, T is of height $O(\log \frac{1}{\varepsilon})$, which implies that the query time is dominated by the value of t. This completes the proof of Theorem 1.2.

7. Preprocessing.

Our principal focus so far has been on establishing the existence of trade-offs between space and query time, without considering how to construct the data structure. In this section we discuss preprocessing issues. We first discuss the preconditioning of K as described in Lemma 2.1 and then discuss the implementation of the access primitives (i)–(iii) needed for SplitReduce as presented at the start of section 3. We assume that the input convex body K is presented as the intersection of a set $\mathcal{H}$ of n halfspaces in $\mathbb{R}^d$. Throughout, let Q denote an arbitrary quadtree cell.

Let t denote the query-time parameter in SplitReduce. As observed in section 3, under our assumption that $t \geq 1$, steps 1 and 2 are not needed, since we can rely entirely on step 3, and therefore access primitives (i) and (ii) are not needed.[2] The remainder of this section will be focused on preconditioning (section 7.1) and the implementation of access primitive (iii), which locally approximates K within Q (section 7.2).

7.1. Preconditioning.

Recall that we assume that K is a (full-dimensional) convex polytope in $\mathbb{R}^d$ that is presented as the intersection of a set of n closed halfspaces. Also recall that Q_0 is the axis-aligned hypercube of unit diameter that is centered at the origin. Our objective is to precondition K by computing an affine transformation that both fattens K and maps it to lie within Q_0. Q_0 has a side length of $1/\sqrt{d}$, and therefore it contains a ball of radius $1/2\sqrt{d}$ centered at the origin. Let B_0 denote this ball, and let r_0 denote its radius. For $0 < \gamma \leq 1$, let γB_0 denote the concentric ball of radius $\gamma r_0 = \gamma/2\sqrt{d}$. We say that a polytope is in γ-*canonical position* if it is nested between γB_0 and B_0 (see Figure 11). Clearly, a polytope that is in canonical position is contained within Q_0 and is γ-fat. The following lemma shows that K can be efficiently mapped into this form, and furthermore an absolute approximation to the transformed body can be easily mapped to a relative approximation of K. (Lemma 2.1 follows as an immediate consequence of this.) Such fattening operations are commonplace in geometric approximation algorithms (see, e.g., [1, 23, 39, 14]), and we employ the standard approach based on minimum enclosing volumes, the John ellipsoid in particular.

[2] If we wished to we could implement access primitive (i) in linear time by linear programming. Also, by testing the membership of each of Q's vertices in K, we could implement a stronger version of access primitive (ii), namely, that of determining whether $Q \subseteq K$ (as opposed to $K \oplus \varepsilon$).

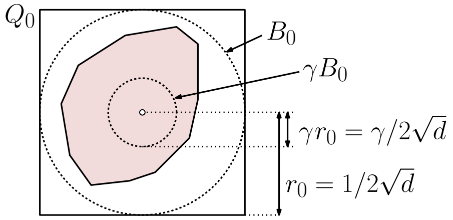


FIG. 11. *A polytope in γ-canonical position.*

LEMMA 7.1. *Let K be a convex polytope in $\mathbb{R}^d$ defined as the intersection of a set $\mathcal{H}$ of n halfspaces, and let $0 < \varepsilon \leq 1$. There is an algorithm that, given $\mathcal{H}$ and ε, in $O(n)$ time computes an affine transformation T that maps K into $(1/d)$-canonical position, such that if P is an absolute $\varepsilon/(d\sqrt{d})$-approximation of $T(K)$, then $T^{-1}(P)$ is a relative ε-approximation of K.*

Proof. Chazelle and Matoušek [26] show that in any fixed dimension, there exists an $O(n)$ time algorithm that, given a convex polytope K presented as the intersection of n halfspaces, computes an ellipsoid E of maximum volume contained within K, also known as the *John ellipsoid* [12]. (At the expense of an increase in the constant factors, we can apply the simpler construction by Barequet and Har-Peled [14].) It is well known from John's theorem (see, e.g., [12]) that K is contained within a uniform scaling of E by a factor of d. It follows from basic linear algebra that the transformation T that maps E to a Euclidean ball $\frac{1}{d}B_0$ achieves the desired result. (Details can be found in [8].) $\square$

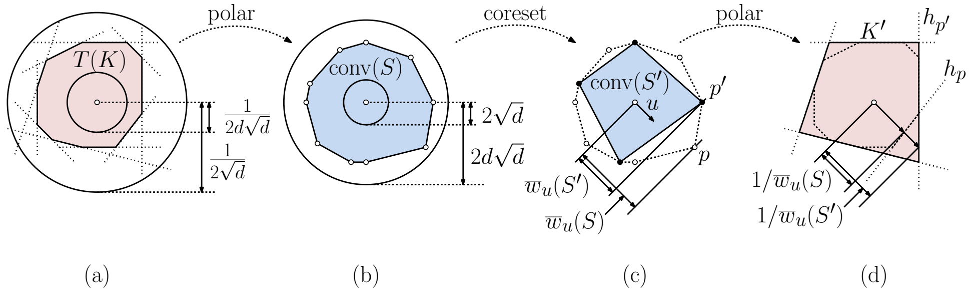
In order to make subsequent processing more efficient, we adapt a standard coreset construction to reduce the number of halfspaces to a function depending only on ε and d. The process will involve some further scaling, which will slightly modify the parameters.

LEMMA 7.2. *Let K be a convex polytope in $\mathbb{R}^d$ defined as the intersection of a set $\mathcal{H}$ of n halfspaces, and let $0 < \varepsilon \leq 1$. There is an algorithm that, given $\mathcal{H}$ and ε, in $O(n + 1/\varepsilon^{d-1})$ time computes an affine transformation T' and a subset $\mathcal{H}' \subseteq \mathcal{H}$ of size $O(1/\varepsilon^{(d-1)/2})$ such that*

(i) *applying T' to the intersection of $\mathcal{H}'$ results in a convex polytope K' that is in $(1/2d)$-canonical position;*
(ii) *furthermore, if P is an absolute $\varepsilon/(4d\sqrt{d})$-approximation of K', then ${T'}^{-1}(P)$ is a relative ε-approximation of K.*

Proof. Given $\mathcal{H}$, we begin by computing the transformation T of Lemma 7.1 in $O(n)$ time. Let $T(K)$ denote the resulting polytope, which is in $(1/d)$-canonical position (see Figure 12(a)). Given a set S of points $\mathbb{R}^d$, the *extent measure* associates each unit vector $u \in \mathbb{R}^d$ with the minimum distance between two hyperplanes orthogonal to u that contain S between them. More formally, define $w_u(S) = \max_{p,q \in S}\langle p - q, u\rangle$ (recalling that $\langle\cdot,\cdot\rangle$ denotes inner product). A subset $S' \subseteq S$ is said to be an *ε-coreset* for the extent measure if for all unit vectors u, $w_u(S') \geq (1-\varepsilon)w_u(S)$. Agarwal, Har-Peled, and Varadarajan [1] showed that, given a set of n points in $\mathbb{R}^d$, it is possible to construct an ε-coreset for the extent measure of size $O(1/\varepsilon^{(d-1)/2})$. Chan presented an algorithm to compute such a coreset in $O(n + (1/\varepsilon)^{d-1})$ time [23].

In order to apply the coreset construction, we first employ the polar dual transformation (recall section 2.3) to $T(K)$, resulting in an n-element point set S of size n



(a) (b) (c) (d)

Fig. 12. *Proof of Lemma* 7.2. *(Not drawn to scale.)*

such that $\mathrm{conv}(S) = \mathrm{polar}(T(K))$ (see Figure 12(b)). It is easy to verify that $\mathrm{conv}(S)$ is nested between an inner ball of radius $2\sqrt{d}$ and an outer ball of radius $2d\sqrt{d}$. Let $\varepsilon' = \varepsilon/4d^2$. We then apply Chan's algorithm to compute an ε'-coreset $S' \subseteq S$ in time $O(n + (1/\varepsilon')^{d-1}) = O(n + (1/\varepsilon)^{d-1})$ (see Figure 12(c)). Let $\mathcal{H}'$ be the subset of $\mathcal{H}$ that results by taking the polar duals of the points of S', and let K' be the convex body that results from intersecting these halfspaces (see Figure 12(d)).

Clearly, $T(K) \subseteq K'$ and $|\mathcal{H}'| = O(1/\varepsilon^{(d-1)/2})$. It follows from a straightforward geometric argument that the Hausdorff distance between $T(K)$ and K' is at most $\varepsilon/2d\sqrt{d}$. (For details, see [8]). Therefore, if P is any absolute $(\varepsilon/2d\sqrt{d})$-approximation to K', then by the triangle inequality (applied to the Hausdorff distance) P is an absolute $(\varepsilon/2d\sqrt{d}) + (\varepsilon/2d\sqrt{d}) = \varepsilon/d\sqrt{d}$ approximation to $T(K)$. By Lemma 7.1, P is a relative ε-approximation of K. We are almost done, but the canonical-position condition fails, because K' need not lie within B_0 of radius $r_0 = 1/2\sqrt{d}$ (even though $T(K)$ does). Since the Hausdorff distance between K' and $T(K)$ is at most $\varepsilon/2d\sqrt{d} \leq 1/2\sqrt{d} = r_0$, K' lies within $2B_0$. The simple fix is to apply a uniform scaling of space by a factor of 1/2 combined with a suitable constant-factor adjustment of ε. The desired conclusion follows as a direct consequence of canonical position. ∎

7.2. Efficient local approximations. Next, we consider the implementation of access primitive (iii), which given a convex body K in γ-canonical position, a quadtree cell Q, and query-time t determines whether there exist t halfspaces whose intersection ε-approximates K within Q. The space and query times stated in Theorem 1.2 are based on the assumption that the number of bounding halfspaces of this local approximating polytope is within a constant factor of optimal. However, we know of no efficient algorithm that can achieve this. In this section we show how to efficiently implement step 3 of SplitReduce approximately in the sense that the number of halfspaces in the approximation exceeds the optimum (for a slightly smaller approximation parameter) by a factor of $O(\log \frac{1}{\varepsilon})$. As shown in Lemma 3.2, this will lead to an increase in the space and query times stated in Theorem 1.2 by a factor of only $O(\log \frac{1}{\varepsilon})$.

A natural approach would be to adapt Clarkson's algorithm for polytope approximation [27]. There are a few messy technical issues involved with such an adaptation. (For example, Clarkson's algorithm applies to the convex hull of a set of points, rather than the intersection of halfspaces.) Since we do not require the strong approximation bounds provided by Clarkson's algorithm, we will instead present a simple direct solution based on a reduction to the set-cover problem. Our approach is to construct a set system where the point set consists of a dense set of points of spacing $\Theta(\varepsilon)$ that covers the portion of Q that is external to $K \oplus c'\varepsilon$ for a suitable constant $c' < 1$.

We associate each bounding halfspace of K with the set of grid points that lie *outside* of this halfspace. We will show that the halfspaces associated with a minimum set cover for this system produce the desired local approximation. We use the greedy set cover heuristic to construct this cover.

Recall that $K \oplus r$ denotes the set of points that lie within Euclidean distance r of K. In order to avoid the complexities of determining whether a point lies outside of $K \oplus c'\varepsilon$, it will suffice for our purposes to perform the simpler test of whether a point lies outside a scaled copy of K. The following lemma follows from a straightforward geometric argument (see [8] for details).

LEMMA 7.3. *For $0 < \gamma \leq 1$ and $0 < \varepsilon \leq 1$, let K be a polytope in $\mathbb{R}^d$ that is in γ-canonical position, and let $K^+ = (1 + 2\sqrt{d}\varepsilon)K$. Then*

$$K \oplus \gamma\varepsilon \;\subseteq\; K^+ \;\subseteq\; K \oplus \varepsilon.$$

While access primitive (iii) does not place any restrictions on the halfspaces used when computing an ε-approximation to K within Q, when the query point q lies outside of K, it may be useful to add further restrictions. In particular, when the query point lies outside of K, it is desirable to obtain a *witness* to nonmembership in the form of a bounding halfspace of K that does not contain q. (This will be exploited in section 9 in the reduction of approximate nearest neighbor searching to approximate polytope membership. The witness hyperplane is used to identify the approximate nearest neighbor.) To achieve this, we would like to use bounding halfspaces from the original polytope in our approximation. By a simple application of Carathéodory's theorem, we can show that we sacrifice only a constant factor by adding this restriction. The following is a straightforward generalization of Lemma 3.1 from Mitchell and Suri [45]. (See [8] for details.)

LEMMA 7.4. *Let K be a convex polytope in $\mathbb{R}^d$ defined as the intersection of a set $\mathcal{H}$ of halfspaces, and let $Q \subseteq Q_0$ be a quadtree cell. If there exists an ε-approximation of K within Q bounded by m halfspaces, then there exists a subset of $\mathcal{H}$ of size at most dm that ε-approximates K within Q.*

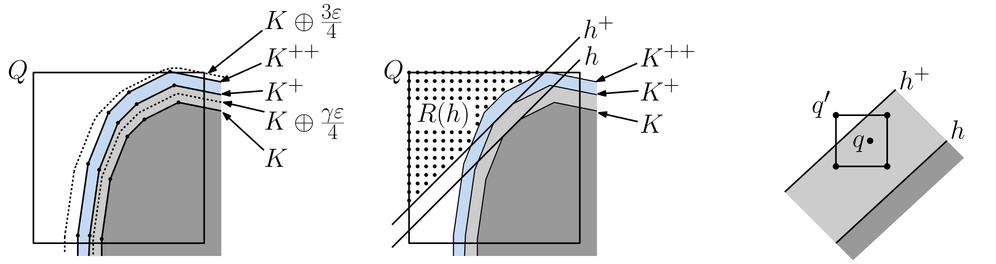
We are now in a position to present our set-cover-based local approximation. This is a bicriteria approximation since it is suboptimal with respect to both the number of bounding halfspaces and the approximation parameter.

LEMMA 7.5. *For $0 < \gamma \leq 1$ and $0 < \varepsilon \leq 1$, let K be a polytope in $\mathbb{R}^d$ in γ-canonical position that is given as the intersection of a set $\mathcal{H}$ of n halfspaces. Let $Q \subseteq Q_0$ be a quadtree cell. In $O(n/\varepsilon^d)$ time, it is possible to compute a subset $\mathcal{H}' \subseteq \mathcal{H}$ such that*

(i) *the intersection of the halfspaces of $\mathcal{H}'$ is an ε-approximation of K within Q,*
(ii) *if m denotes the minimum number of halfspaces needed to $(\gamma\varepsilon/2)$-approximate K within Q, then $|\mathcal{H}'|$ is $O(m \log \frac{1}{\varepsilon})$.*

Proof. First, we may assume without loss of generality that $\varepsilon \leq 2/\sqrt{d}$. Otherwise, setting $\varepsilon = 2/\sqrt{d}$ will certainly satisfy (i) and will only affect the constant factors in the asymptotic bounds of claim (ii) and the construction time. Define $\beta = \sqrt{d}\varepsilon/2$. By the above assumption, we have

$$(1 + \beta)^2 \;=\; \left(1 + \sqrt{d}\varepsilon + \frac{d\varepsilon^2}{4}\right) \;\leq\; 1 + \frac{3}{2}\sqrt{d}\varepsilon.$$



(a) (b) (c)

FIG. 13. *Proof of Lemma* 7.5.

Let $K^{+} = (1+\beta)K$ and let $K^{++} = (1+\beta)K^{+} = (1+\beta)^2 K$. By applying Lemma 7.3 but with ε taking on the values $\varepsilon/4$ and $3\varepsilon/4$, respectively, we have

$$K \oplus \frac{\gamma\varepsilon}{4} \;\subseteq\; K^{+} \;\subseteq\; K^{++} \;\subseteq\; K \oplus \frac{3\varepsilon}{4} \tag{7}$$

(see Figure 13(a)). Let $\delta = \gamma\varepsilon/4$ and let G denote the vertices of a hypercube grid of diameter δ. Let R be the set of grid points that lie within Q but outside of K^{++}, that is, $R = G \cap (Q \setminus K^{++})$. Since $Q \subseteq Q_0$, the resulting set is of size $O(1/\varepsilon^d)$, and hence it can be computed in time $O(1/\varepsilon^d) \cdot |\mathcal{H}| = O(n/\varepsilon^d)$, by testing each grid point against each halfspace of $\mathcal{H}$. Because $\gamma \le 1$, we have $\delta \le \varepsilon/4$.

Next, we define a set system to model the approximation process. For each $h \in \mathcal{H}$ we define a subset $R(h) \subseteq R$ as follows. First, let $h^{+} = (1+\beta)h$ denote the corresponding bounding halfspace of the scaled body K^{+} (see Figure 13(b)). Define $R(h)$ to be the subset of points of R that lie outside of h^{+}. Consider a set system consisting of the points of R and the sets $R(h)$ for all $h \in \mathcal{H}$. Since every point of R lies outside of K^{++}, and hence outside of K^{+}, together these sets cover R. The resulting collection of sets has total cardinality $O(n/\varepsilon^d)$.

Consider any set cover C of the resulting set system. Let $P(C)$ denote the polyhedron that results by intersecting the halfspaces h whose associated set $R(h)$ is included in this cover. (Note that the sets $R(h)$ are based on the halfspaces bounding the scaled body K^{+}, while $P(C)$ is based on the halfspaces bounding the original body K.) We assert that $P(C)$ ε-approximates K within Q. It suffices to show that for any point $q \in Q \setminus (K \oplus \varepsilon)$, q is not in $P(C)$. First, observe that for such a point q, all the vertices of the grid cell in which it lies are within distance δ of q. Therefore, by the triangle inequality, each such vertex is at distance at least $\varepsilon - \delta \ge 3\varepsilon/4$ from K. Since by (7), $K^{++} \subseteq K \oplus 3\varepsilon/4$, these vertices are all exterior to K^{++}, which implies that they are all members of R. Let q' be any of these vertices. Since C is a cover, there exists a halfspace $h \in \mathcal{H}$ such that $R(h)$ is in the cover and contains this point. This implies that q' lies outside the associated halfspace h^{+} (see Figure 13(c)). Because K is in γ-canonical position, the minimum distance between h's bounding hyperplane and the origin is at least $\gamma/2\sqrt{d}$. Therefore the distance between any point in h to any point exterior to h^{+} is at least

$$\frac{\gamma}{2\sqrt{d}}((1+\beta)-1) \;=\; \frac{\gamma}{2\sqrt{d}} \cdot \frac{\sqrt{d}\varepsilon}{2} \;=\; \frac{\gamma\varepsilon}{4} \;=\; \delta.$$

It follows by the triangle inequality that q is exterior to h, and therefore it lies outside of $P(C)$, as desired.

Let C' denote a set cover that results by running the greedy heuristic [29] on the aforementioned set system. By standard results on the greedy heuristic, the size of the resulting cover exceeds that of an optimal cover by a factor of at most $\ln |R| = O(\log \frac{1}{\varepsilon})$. C' can be computed in time that is proportional to the total cardinality of the sets of the set system, which is $O(n/\varepsilon^d)$. Let $\mathcal{H}'$ denote the associated set of halfspaces, and let $P(C')$ denote the intersection of these halfspaces. By the above remarks, $P(C')$ is an ε-approximation to K within Q, which establishes claim (i).

To establish (ii), consider a $(\gamma\varepsilon/4)$-approximation of K within Q that is bounded by the minimum number m of halfspaces. By Lemma 7.4 there exists such an approximation that uses only the bounding halfspaces of K, such that the number of halfspaces is larger by a factor of at most d. Let P^+ denote this approximation, and let $\mathcal{H}^+ \subseteq \mathcal{H}$ denote its bounding halfspaces. By (7), we have $P^+ \subseteq K \oplus \gamma\varepsilon/4 \subseteq K^+$. Let $P^{++} = (1+\beta)P^+$. Clearly, $P^{++} \subseteq (1+\beta)K^+ = K^{++}$. Therefore, every point of R lies outside of P^{++}. It follows that the sets $R(h)$ associated with the halfspaces h that bound P^+ form a set cover of R within our system. Letting C^{++} denote this cover, we have $|C'| \leq O(\log \frac{1}{\varepsilon}) \cdot |C^{++}| \leq O(\log \frac{1}{\varepsilon}) \cdot dm = O(m \log \frac{1}{\varepsilon})$, as desired. ∎

We can now present the main result of this section, which summarizes the preprocessing time.

Lemma 7.6. *Given a full-dimensional convex polytope K in $\mathbb{R}^d$ defined as the intersection of a set of n halfspaces, approximation parameter $0 < \varepsilon \leq 1$, and query time parameter $t \geq 1$, there is an algorithm that runs in time $O(n + 1/\varepsilon^{c_p d})$ for some constant c_p (which does not depend on d) that constructs a data structure satisfying Theorem 1.2 but with an additional factor of $O(\log \frac{1}{\varepsilon})$ in both the space and query times.*

Proof. Given K's bounding halfspaces, we apply Lemma 7.2. In $O(n + 1/\varepsilon^{d-1})$ time we obtain a polytope K', such that K' is in γ-canonical position for $\gamma = 1/2d$. K' is bounded by a subset $\mathcal{H}'$ of halfspaces of size $n' = O(1/\varepsilon^{(d-1)/2})$, and the problem of computing a relative ε-approximation of K reduces to the problem of computing an absolute ε'-approximation of K', where $\varepsilon' = \varepsilon/4d\sqrt{d}$.

Ideally, we would like to invoke SplitReduce on K' using ε' as the approximation parameter and t as the query time parameter. Since we do not know how to determine minimum-sized convex approximations efficiently, we will need to relax our expectations. For any quadtree cell Q generated by SplitReduce, we apply Lemma 7.5 on the set $\mathcal{H}'$ of halfspaces. By claim (i) of this lemma, after $O(n'/(\varepsilon')^d) = O(1/\varepsilon^{3d/2})$ time, a subset $\mathcal{H}'' \subseteq \mathcal{H}'$ can be computed that is an ε'-approximation of K' within Q. Irrespective of the choice of the query time, the maximum number of quadtree cells generated by SplitReduce is $O(1/\varepsilon^d)$, and therefore (after preconditioning) the overall running time of SplitReduce is $O(1/\varepsilon^{5d/2})$. Combined with the $O(n + 1/\varepsilon^d)$ time for preconditioning, the algorithm's overall running time is $O(n+1/\varepsilon^{c_p d})$, where $c_p = 5/2$.

Let $\varepsilon'' = \gamma\varepsilon'/2 = \gamma\varepsilon/8d\sqrt{d}$. By Lemma 7.5(ii) the number of halfspaces in $\mathcal{H}''$ is within a factor of $\rho = O(\log \frac{1}{\varepsilon'}) = O(\log \frac{1}{\varepsilon})$ of the size of the minimum-sized ε''-approximation of K' within Q. Since $\varepsilon'' = \beta\varepsilon'$ for a constant β, Lemma 3.2 implies that the conclusions of Theorem 1.2 hold but with an additional factor of $\rho = O(\log \frac{1}{\varepsilon})$ in both the space and query times. ∎

8. Lower bound.

In this section, we establish lower bounds on the space-time trade-offs obtained by SplitReduce for polytope membership. In particular, we will prove Theorem 1.3. Our approach is similar to the lower bound proof of [10]. (Note that this is a lower bound on the performance of SplitReduce, not on the problem complexity. It applies to the stronger existential version of the algorithm.) It is based on analyzing the performance of the algorithm on a particular convex body, a generalized hypercylinder that is curved in $k+1$ dimensions and flat in $d-1-k$ dimensions. We select the value of k that produces the best lower bound on the storage as a function of t, ε, and d. Throughout, we use the term ε-approximation in the absolute sense, as defined in section 2.1.

As mentioned earlier, it is well known that $\Omega(1/\varepsilon^{(d-1)/2})$ facets are required to ε-approximate a Euclidean ball of unit radius (see, e.g., [18]), and this holds for any polytope that is sufficiently close to a ball in terms of Hausdorff distance. The following utility lemma generalizes this observation to different diameters. The proof is a straightforward exercise in geometry. (Details can be found in [8].)

Lemma 8.1. *Let ε and Δ be real parameters, where $0 < \varepsilon \le \Delta/4$. There exists a constant c_b and a polytope P in $\mathbb{R}^d$ of diameter at most Δ such that any outer ε-approximation of P requires at least $c_b(\Delta/\varepsilon)^{(d-1)/2}$ facets.*

Intuitively, in order to produce a polytope that is hard to approximate, it should have high curvature. If the curvature is high in all dimensions, however, the polytope will have a small surface area, and this will make it easier to approximate. Our approach is to consider polytopes based on generalized cylinders, which have constant curvature in some dimensions but are flat in others. Our next lemma introduces such a cylindrical polytope where the number of curved dimensions has been carefully chosen to maximize the space needed by our algorithm for a given query time. Theorem 1.3 is an immediate consequence.

Lemma 8.2. *There exists a polytope P in $\mathbb{R}^d$ such that for all sufficiently small positive ε (depending on d and α) and $t = 1/\varepsilon^{(d-1)/\alpha}$, the output of SplitReduce(K, Q_0) on P has total space*

$$\Omega\left(1/\varepsilon^{(d-1)\left(1-\frac{2\sqrt{2\alpha}-3}{\alpha}\right)-1}\right).$$

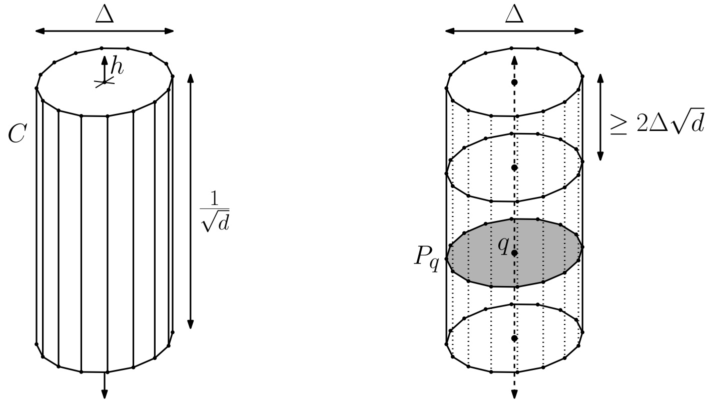
Proof. To start, as a function of α, we wish to compute an integer dimension k in order to apply Lemma 8.1. Define reals $\delta = \sqrt{\alpha/2}/(d-1)$, $\kappa = (d-1)\sqrt{2/\alpha}$, and $\kappa' = \kappa(1+\delta)$. We observe first that

$$\kappa' - \kappa \;=\; \delta(d-1)\sqrt{2/\alpha} \;=\; 1.$$

Let $k = \lceil \kappa \rceil$, implying that $\kappa \le k \le \kappa'$. (Although we do not include the derivation here, κ has been chosen to produce the best lower bound, but since it is not necessarily an integer, k is obtained by rounding to a nearby integer.) Since $\alpha \ge 4$ and $d \ge 2$, we have $1 \le k \le d-1$.

Let c_b denote the constant of Lemma 8.1, and let $\Delta = \varepsilon((2^d+1)t/c_b)^{2/k}$. By our assumptions about d and α, we have $t = 1/\varepsilon^{\Theta(1)}$ and $\Delta = \varepsilon \cdot t^{\Theta(1)}$. It follows that for all sufficiently small ε, $\Delta/4 \ge \varepsilon$. Let h denote the linear subspace spanned by the first $k+1$ coordinate axes. We apply Lemma 8.1 in $\mathbb{R}^{k+1}$ for this value of Δ. The resulting polytope P (lying in h) has the property that the number of facets of any ε-approximation is at least

$$c_b\left(\frac{\Delta}{\varepsilon}\right)^{k/2} \;=\; c_b\left(\frac{\varepsilon\left(\frac{(2^d+1)t}{c_b}\right)^{2/k}}{\varepsilon}\right)^{k/2} \;=\; (2^d+1)t.$$



(a) (b)

Fig. 14. *Lemma* 8.2 *for* $d = 3$ *and* $k = 2$.

We can bound P's diameter by observing that for all sufficiently small ε

$$\operatorname{diam}(P) \leq \Delta \ = \ \varepsilon\left(\frac{(2^d+1)t}{c_b}\right)^{2/k} \ \leq \ \varepsilon\left(\frac{2^d+1}{c_b \cdot \varepsilon^{(d-1)/\alpha}}\right)^{2/\kappa}$$
$$= \varepsilon\left(\frac{2^d+1}{c_b \cdot \varepsilon^{(d-1)/\alpha}}\right)^{\sqrt{2\alpha}/(d-1)}.$$

(Here we made use of the fact that for all sufficiently small ε, the quantity raised to power of $2/k$ is greater than 1.) Letting $c_b' = ((2^d+1)/c_b)^{\sqrt{2\alpha}/(d-1)}$, we obtain

$$\operatorname{diam}(P) \ \leq \ c_b'\varepsilon\left(\frac{1}{\varepsilon^{(d-1)/\alpha}}\right)^{\sqrt{2\alpha}/(d-1)} \ = \ c_b'\varepsilon^{1-\sqrt{2/\alpha}}.$$

Since $\alpha \geq 4$, for all sufficiently small ε, we have $\operatorname{diam}(P) \leq 1/\sqrt{d}$. Therefore, P can be enclosed within $Q_0^{(k+1)}$.

Returning to $\mathbb{R}^d$, consider an infinite polyhedral hypercylinder whose "axis" is the $(d-1-k)$-dimensional orthogonal complement of h and whose "cross section" (i.e., intersection with any $(k+1)$-dimensional hyperplane parallel to h) is P. Define the polytope C to be the truncated cylinder obtained by intersecting the infinite hypercylinder with hypercube $Q_0^{(d)}$ (see Figure 14(a)). Let T denote the output of SplitReduce$(K, Q_0^{(d)})$ for C, ε, and t. We will show that T's total space satisfies the bound given in the lemma's statement. To do this, let Σ denote any set of points placed on C's axis such that the distance between each pair of points is at least $2\Delta\sqrt{d}$. (In the degenerate case where $k = d-1$ the axis is 0-dimensional and Σ degenerates to a single point.) By a simple packing argument, there exists such a set having $\Omega(1/\Delta^{d-1-k})$ points.

For any $q \in \Sigma$, let P_q denote the cross-section of C passing through q (see Figure 14(b)). Consider the set of leaf cells of T that intersect P_q. By applying Lemma 8.1 to the $(k+1)$-dimensional hyperplane on which P lies, it follows that these cells together must contain at least $(2^d+1)t$ halfspaces. We count the contributions of these cells by classifying them into two types. We say that a leaf cell of T is *large* if its side length is at least Δ, and otherwise it is *small*. By a simple packing argument, the number of large leaf cells intersecting P_q is at most 2^d. Since each leaf cell contains at most t halfspaces, the large leaf cells can together contain at most $2^d t$ halfspaces.

Therefore, the small leaf cells intersecting P_q together contain at least $(2^d + 1)t - 2^d t = t$ halfspaces. Because the points of Σ are separated from each other by distance at least $2\Delta\sqrt{d}$, which is strictly larger than the diameter of any small leaf cell, each small leaf cell can intersect P_q for at most one $q \in \Sigma$. Therefore, the total space contribution of all the small leaf cells for all points of Σ is at least $t \cdot |\Sigma|$. Let $c''_b = (c_b/(2^d + 1))^{2(d-1-k)/k}$. T's total space can be asymptotically bounded from below as

$$\frac{t}{\Delta^{d-1-k}} \;=\; \frac{t}{\left(\varepsilon\left(\frac{(2^d+1)t}{c_b}\right)^{2/k}\right)^{d-1-k}} \;=\; \frac{c''_b \cdot t}{\left(\varepsilon \cdot t^{2/k}\right)^{d-1-k}} \;=\; \frac{c''_b \cdot t^{1-2(d-1-k)/k}}{\varepsilon^{d-1-k}}.$$

Clearly, $c''_b = \Theta(1)$. Recall that $t = 1/\varepsilon^{(d-1)/\alpha}$. Then, T's total space is asymptotically bounded from below as

$$\left(\frac{1}{\varepsilon}\right)^{(d-1)-k+\frac{d-1}{\alpha}\left(1-\frac{2(d-1-k)}{k}\right)} \;=\; \left(\frac{1}{\varepsilon}\right)^{(d-1)-k+\frac{d-1}{\alpha}\left(3-\frac{2(d-1)}{k}\right)}. \tag{8}$$

Let $E(\alpha)$ denote this exponent. In order to complete the proof, we provide a lower bound on $E(\alpha)$. We use the fact that $\kappa \le k \le \kappa'$, apply the definitions of κ, κ', and δ, and apply straightforward manipulations to obtain

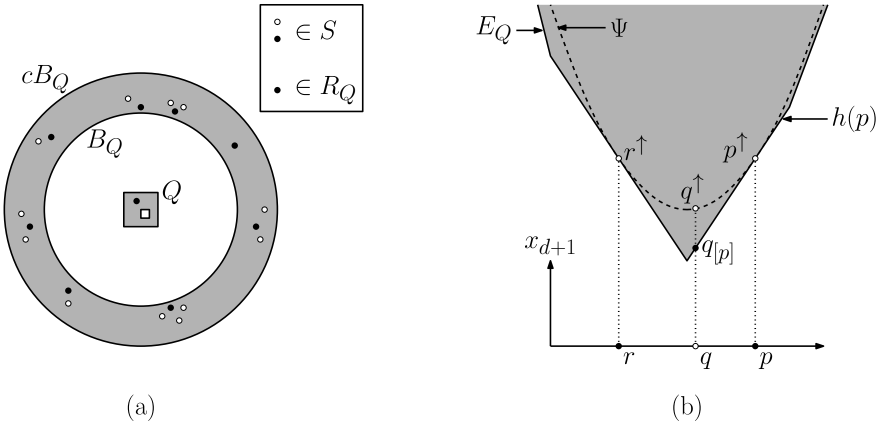
$$E(\alpha) \ge (d-1) - \kappa' + \frac{d-1}{\alpha}\left(3 - \frac{2(d-1)}{\kappa}\right) \;=\; (d-1)\left(1 - \frac{2\sqrt{2\alpha}-3}{\alpha}\right) - 1.$$

Substituting this value for the exponent in (8) completes the proof. □

9. Approximate nearest neighbor searching. In this section, we present a reduction from approximate nearest neighbor searching to approximate polytope membership, which will allow us to prove Theorem 1.4. Our reduction will involve the following additional assumptions regarding the implementation of SplitReduce. First (as in section 7), we assume that K is presented as the intersection of n halfspaces. Second, we assume that a leaf node is labeled as "inside" only if it lies entirely within K (as opposed to lying within $K \oplus \varepsilon$ as described in SplitReduce). Third, we assume that leaf cells that store halfspaces use only bounding halfspaces of K.

Clearly, these assumptions do not affect the data structure's correctness. We assert that they do not affect the data structure's asymptotic query time or space bounds. Regarding the second assumption, observe that for any cell Q that lies within $K \oplus \varepsilon$, K can be ε-approximated within Q using a single halfspace (any halfspace that contains Q suffices). Regarding the third assumption, recall that Lemma 7.4 shows that we may assume that the approximating halfspaces for each node are drawn from the input halfspaces at the expense of a constant factor increase in the query time.

The reduction from approximate nearest neighbor searching to approximate polytope membership is based on the AVD construction from [10]. The AVD employs a height balanced variant of a quadtree, a balanced box decomposition (BBD) tree [11] to be precise. Each cell of a BBD tree corresponds to the set theoretic difference of two quadtree cells, an *outer box* and an optional *inner box*. Each leaf cell of the tree stores a set of *representative points* with the property that for any query point q lying



(a) (b)

FIG. 15. *Approximate nearest neighbor searching:* (a) *Lemma* 9.1 *(black points are members of R_Q),* (b) *the lifting transformation. (Note that the figure is not drawn to scale, and the paraboloid in* (b) *has been translated to aid legibility.)*

within this cell, at least one of these representatives is an ε-nearest neighbor of q. A query is answered by locating the leaf cell that contains the query point and then determining the nearest representative from this cell (by brute force). The AVD's space is dominated by the total number of representatives over all the leaf cells. The query time is the height of the tree plus the number of representatives in the leaf cell. A data structure for nearest neighbor searching is said to be in the *AVD model* if it has this general form, that is, a covering of the query region by hyperrectangles of bounded aspect ratio, each of which is associated with a set of representative points [10]. Lower bounds on the performance of any data structure in the AVD model were given in [10].

The reader need not be familiar with the details of the AVD data structure. The next lemma encapsulates the important technical information needed for our reduction. It follows easily from the proofs of Lemmas 6.1 and 8.1 in [10]. Given a cell Q in a BBD tree, let B_Q denote the ball of radius $2 \cdot \mathrm{diam}(Q)$ whose center coincides with the center of Q's outer box (see Figure 15(a)). Given a Euclidean ball B of radius r and positive c, let cB denote the ball concentric with B of radius cr.

LEMMA 9.1. *Let $0 < \varepsilon \leq 1/2$ be a real parameter and X be a set of n points in $\mathbb{R}^d$. It is possible to construct a BBD tree T with $O(n \cdot \log \frac{1}{\varepsilon})$ nodes, where each leaf cell Q stores a subset $R_Q \subset X$ satisfying the following properties:*

(i) *For any point q in Q, one of the points in R_Q is an ε-approximate nearest neighbor of q.*
(ii) *At most one point of R_Q is contained in the ball B_Q, and the remaining points of R_Q are contained in $c_q B_Q \setminus B_Q$ for some constant c_q (which depends on the dimension).*
(iii) *The total number of representative points over all the leaf cells of T is $O(n \cdot \log \frac{1}{\varepsilon})$.*

Moreover, it is possible to compute the tree T and the sets R_Q for all the leaf cells in total time $O(n \cdot \log n \cdot \log \frac{1}{\varepsilon})$, and the cell that contains a query point can be located in time $O(\log n + \log\log \frac{1}{\varepsilon})$.

In the AVD data structure of [10] the closest representative point to a query point is determined by brute-force enumeration of the elements of R_Q. We consider whether it is possible search them more efficiently by reduction to polytope approximation.

The following lemma explains how to connect Lemma 9.1 with approximate polytope membership queries. Our construction uses the well-known *lifting transformation* [3, 34]. Let $(x_1, \ldots, x_{d+1})$ denote the coordinates of $\mathbb{R}^{d+1}$, and let us think of the $(d+1)$st coordinate axis as being directed vertically upward. Let Ψ denote the paraboloid $x_{d+1} = \sum_{i=1}^{d} x_i^2$. Given a point $p \in \mathbb{R}^d$, let $p^{\uparrow}$ denote the vertical projection of p onto Ψ (see Figure 15(b)), and let $h(p)$ denote the hyperplane tangent to Ψ at $p^{\uparrow}$. That is, the points of $h(p)$ satisfy $x_{d+1} = \sum_{i=1}^{d} 2p_i x_i - \|p\|^2$. Given $q \in \mathbb{R}^d$, let $q_{[p]}$ denote the point on $h(p)$ hit by a vertical ray shot downward from $q^{\uparrow}$. A straightforward consequence of the definition of Ψ is that the squared distance between q and p in $\mathbb{R}^d$ is equal to the length of this vertical segment, that is, $\|qp\|^2 = \|q^{\uparrow} q_{[p]}\|$.

This suggests the following approach to computing the closest representative point through vertical ray shooting. Consider the (unbounded) convex polyhedron that results by taking the upper envelope of the hyperplanes $h(p)$ associated with the lifted representatives. Given the query point $q \in \mathbb{R}^d$, a ray shot vertically downward from $q^{\uparrow}$ hits some facet of this polyhedron. It follows from the above remarks that the representative associated with this hyperplane is the closest to q. We can simulate ray shooting by applying polytope membership queries in concert with binary search. Of course, some care will be needed to map this problem into our context, which assumes a bounded polytope and approximation.

Lemma 9.2. *Let $0 < \varepsilon \leq 1/2$ be a real parameter and consider a quadtree cell Q and a set of representative points R_Q as in Lemma* 9.1. *Given a data structure for ε-approximate polytope membership in d-dimensional space with query time $t_d(\varepsilon)$ and space $s_d(\varepsilon)$, it is possible to preprocess R_Q into an approximate nearest neighbor data structure for query points in Q with query time $O(t_{d+1}(\varepsilon) \cdot \log \frac{1}{\varepsilon})$ and space $O(s_{d+1}(\varepsilon))$.*

Proof. Since at most one point of R_Q is contained in B_Q, the corresponding point may be inspected separately without increasing the complexity bounds. Therefore, we may assume that all points of R_Q are contained in $c_q B_Q \setminus B_Q$.

Although we assume that the errors in polytope membership are absolute (because of standardization), errors in approximate nearest neighbor searching are relative. That is, a point r is an ε-approximate nearest neighbor of q if $\|qr\| \leq (1+\varepsilon)\|qp\|$, where p is q's true nearest neighbor. Because errors are relative, we may assume that space has been translated and uniformly scaled so that Q is mapped to $Q_0^{(d)}$, the hypercube of unit diameter centered at the origin in $\mathbb{R}^d$. As a result, B_Q is mapped to a ball of radius 2. It follows that the distance from any point of Q to any point of R_Q is greater than 1. Therefore, an absolute error of ε implies a relative error of at most ε.

In order to reduce nearest neighbor searching among the points of R_Q to polytope membership, let E_Q denote the upper envelope, that is, the intersection of the upper halfspaces, of the hyperplanes $h(p)$, for all $p \in R_Q$ (the shaded region in Figure 15(b)). As mentioned above, the facet of E_Q hit by shooting a ray vertically downward from $q^{\uparrow}$ corresponds to the closest point of R_Q to q.

Since the upper envelope is unbounded, we first compute a bounded convex polytope on which to perform approximate membership queries. Because the query points lie in Q, we are only interested in the portion of E_Q that projects vertically onto Q. Given that the distance of any point $p \in R_Q$ to the origin is at most $2c_q = O(1)$, it follows that the portion of E_Q of interest fits within an axis-aligned $(d+1)$-dimensional hypercube of constant diameter that is centered at the origin. Let Q' denote such a hypercube, let $K_Q = E_Q \cap Q'$, and let $\varepsilon' = \varepsilon/6c_q$. We invoke SplitReduce to construct an ε'-approximate membership data structure for K_Q. (More formally, we first scale Q' into standard form, and we scale ε' by the same factor. We then apply SplitReduce

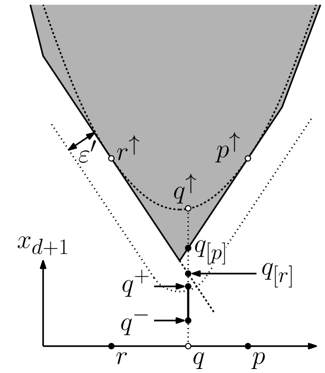


FIG. 16. *Proof of Lemma* 9.2. *(Not drawn to scale.)*

with the scaled value of ε'. Since Q' is of constant diameter, the scale factor will also be a constant, and therefore only the constant factors in the analysis will be affected. We then apply an inverse scaling to obtain the desired ε'-approximating polytope for K_Q.)

We simulate the ray shooting process by a binary search to locate the contact point approximately. Consider the vertical segment formed by intersecting Q' with the vertical line passing through $q^{\uparrow}$. The upper endpoint of this segment is clearly inside K_Q and its lower endpoint is outside. We repeatedly split the segment at its midpoint, perform an approximate polytope membership query, and retain the subsegment whose upper endpoint is (approximately) inside K_Q and whose lower endpoint is (approximately) outside. We terminate the search when the length of the segment falls below ε'. Since Q' is of constant diameter, the search terminates after $O(\log \frac{1}{\varepsilon})$ membership queries. Let us denote the endpoints of this final segment as q^{+} (upper) and q^{-} (lower).

Recall our assumption that cells are labeled by SplitReduce as "inside" or "outside" only if they lie entirely inside or outside K_Q, respectively. It follows that as we traverse the cells that intersect the segment $q^{+}q^{-}$ from top to bottom, we cannot transition directly from an "inside" cell to an "outside" cell. Therefore, at least one of these cells must contain a set of representative hyperplanes. Let $h(r)$ denote the hyperplane having the topmost intersection with the vertical ray. We return r as the approximate nearest neighbor (see Figure 16). It is easy to see that this algorithm satisfies the desired time and space bounds.

All that remains is to establish correctness, by showing that r is indeed an ε-approximate nearest neighbor of q. In order to do this, let p be q's true nearest neighbor in R_Q. Due to the nature of the binary search, q^{+} lies within distance ε' of K_Q. (Note that it might lie within K_Q.) Thus, the distance from q^{+} to the upper halfspace bounded by $h(p)$ is at most ε'. By the triangle inequality, the distance from q^{-} to this halfspace is at most $\varepsilon' + \varepsilon' = 2\varepsilon'$. Since p is q's true nearest neighbor, $q_{[p]}$ lies on ∂K_Q, and so the hyperplane $h(p)$ separates q^{-} from K_Q. This implies that the distance from q^{-} to $h(p)$ is also not greater than $2\varepsilon'$.

We claim that the vertical distance from q^{-} to $q_{[p]}$ is at most ε. To see why, recall that p lies within a ball of radius $2c_q$ centered at the origin. This implies that $h(p)$ cannot be too steep, that is, the angle formed between $h(p)$'s normal vector and the vertical axis can be bounded away from $\pi/2$ by a constant. By basic linear algebra, it can be shown that the ratio of the vertical and orthogonal distances of any point to $h(p)$ is bounded above by $\sqrt{4c_q^2+1<3c_q}$. Therefore, we have $\|q_{[p]}q^{-}\| \leq 3\,c_q\,(2\varepsilon') = \varepsilon$, as desired.

Because r is the witness produced by the algorithm, $h(r)$ separates q^- from K_Q, which implies that $q_{[r]}$ lies above q^-. Thus, we have $\|q_{[p]}q_{[r]}\| \leq \|q_{[p]}q^-\| \leq \varepsilon$. Therefore,

$$\|qr\|^2 \;=\; \|q^\uparrow q_{[r]}\| \;=\; \|q^\uparrow q_{[p]}\| + \|q_{[p]}q_{[r]}\| \;\leq\; \|q^\uparrow q_{[p]}\| + \varepsilon.$$

By the lifting transformation, we have $\|q^\uparrow q_{[p]}\| = \|qp\|^2$, and combining this with the fact that $\|qp\| \geq 1$, we have

$$\|qr\|^2 \;\leq\; \|qp\|^2 + \varepsilon \;\leq\; \|qp\|^2 + \|qp\|^2\varepsilon \;=\; \|qp\|^2(1+\varepsilon) \;\leq\; (\|qp\|(1+\varepsilon))^2.$$

Therefore, r is an ε-approximate nearest neighbor of p, which completes the proof. □

The above lemma shows how to apply approximate polytope membership to efficiently answer approximate nearest neighbor queries within each cell of the AVD. To obtain a complete data structure for approximate nearest neighbor searching we apply this to every leaf cell of the AVD.

LEMMA 9.3. *Let $0 < \varepsilon \leq 1/2$ be a real parameter and X be a set of n points in $\mathbb{R}^d$. Given a data structure for approximate polytope membership in d-dimensional space with query time at most $t_d(\varepsilon)$ and storage $s_d(\varepsilon)$, it is possible to preprocess X into an approximate nearest neighbor searching data structure with query time $O(\log n + t_{d+1}(\varepsilon) \cdot \log \frac{1}{\varepsilon})$ and space*

$$O\left(n \log \frac{1}{\varepsilon} + n\, \frac{s_{d+1}(\varepsilon)}{t_{d+1}(\varepsilon)}\right).$$

Proof. Following Lemma 9.1, construct a BBD-tree T, and for each leaf cell Q of T, construct the set of representative points R_Q. For each leaf cell such that $|R_Q| \leq t_{d+1}(\varepsilon) \cdot \lg \frac{1}{\varepsilon}$, simply store the set R_Q and answer the corresponding queries by brute force. For the nodes with $|R_Q| > t_{d+1}(\varepsilon) \cdot \lg \frac{1}{\varepsilon}$, use the construction from Lemma 9.2.

To answer an approximate nearest neighbor query we search the AVD of Lemma 9.1 to find the leaf cell containing the query point and then apply Lemma 9.2. Thus, the query time is

$$O\left(\log n + \log\log \frac{1}{\varepsilon} + t_{d+1}(\varepsilon) \cdot \log \frac{1}{\varepsilon}\right) \;=\; O\left(\log n + t_{d+1}(\varepsilon) \cdot \log \frac{1}{\varepsilon}\right).$$

To bound the total space, observe from Lemma 9.1(iii) that the total number of representative points is $O(n \log \frac{1}{\varepsilon})$. Thus, by a simple counting argument, the number of leaf cells with more than $t_{d+1}(\varepsilon) \cdot \lg \frac{1}{\varepsilon}$ representatives is $O(n/t_{d+1}(\varepsilon))$. Therefore, the total space of the data structure is $O(n \log \frac{1}{\varepsilon} + n(s_{d+1}(\varepsilon)/t_{d+1}(\varepsilon)))$. □

Because of its reliance on binary search, the generic reduction given in Lemmas 9.2 and 9.3 is not formally in the AVD model. Recall that the AVD model is important because lower bounds have been established in this model [10], and thus these lower bounds do not apply here. However, by sacrificing generality and a factor of $O(\log \frac{1}{\varepsilon})$ in the space bound, we can exploit the properties of SplitReduce to obtain a data structure that is in the AVD model.

LEMMA 9.4. *Let $0 < \varepsilon \leq 1/2$ be a real parameter and X be a set of n points in $\mathbb{R}^d$. Given a split-reduce data structure for approximate polytope membership in*

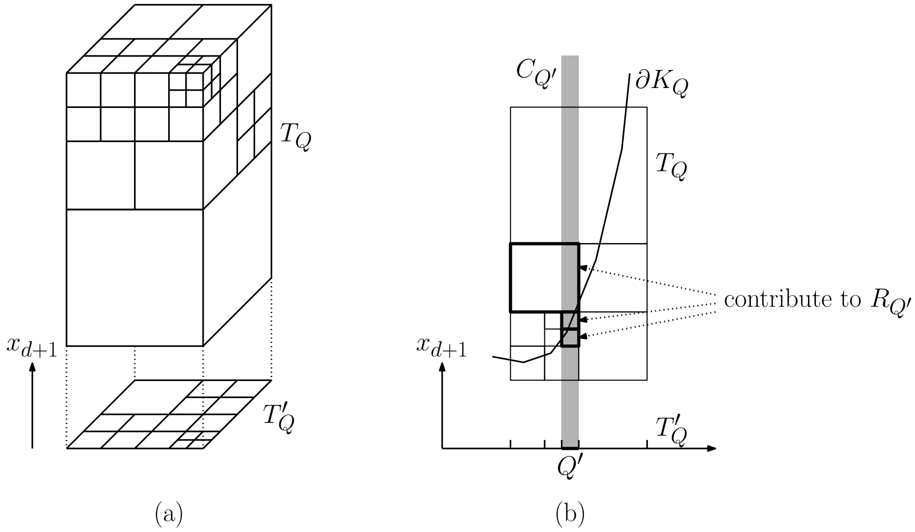


FIG. 17. *Producing an approximate nearest neighbor data structure in the AVD model.*

d-dimensional space with query time at most $t_d(\varepsilon)$ and storage $s_d(\varepsilon)$, it is possible to preprocess X into an approximate nearest neighbor data structure in the AVD model with query time $O(\log n + t_{d+1}(\varepsilon) \cdot \log \frac{1}{\varepsilon})$ and space

$$O\left(n\left(1+\frac{s_{d+1}(\varepsilon)}{t_{d+1}(\varepsilon)}\right)\log\frac{1}{\varepsilon}\right).$$

Proof. As in Lemma 9.3, construct a BBD-tree T, and for each leaf cell Q of T, construct the set of representative points R_Q. We may assume that $|R_Q| > t_{d+1}(\varepsilon) \cdot \lg \frac{1}{\varepsilon}$, since otherwise we just use the points of R_Q as the representatives. In order to handle query points lying within Q, we apply Lemma 9.2, where queries are answered using the tree produced by SplitReduce. Let T_Q denote the resulting tree. We exploit the fact that the SplitReduce data structure associates a collection of hyperplanes with each leaf cell of T_Q, and by the nature of our reduction, each of these hyperplanes corresponds to a lifted point of R_Q. These lifted points will play the role of nearest neighbor representatives. Intuitively, our approach is to "undo" the lifting transformation by projecting the leaf cells of T_Q vertically from $\mathbb{R}^{d+1}$ down to $\mathbb{R}^d$ and then building a d-dimensional AVD structure based on this projection.

The projection of the cells of T_Q onto $\mathbb{R}^d$ naturally defines a quadtree subdivision of $\mathbb{R}^d$, which we denote by T'_Q (see Figure 17(a)). For each leaf cell Q' of T'_Q, let $C_{Q'}$ denote the infinite vertical cylinder in $\mathbb{R}^{d+1}$ whose cross section is Q' (see Figure 17(b)). Because Q' is a leaf, any leaf cell of T_Q that intersects this cylinder projects onto a hypercube that contains Q'.

Recall the lifted polytope K_Q of Lemma 9.2. For each leaf cell of T_Q that contains a point whose vertical distance from ∂K_Q is at most ε, we create a representative point corresponding to each of the hyperplanes that SplitReduce associates with this leaf cell. We denote the resulting collection of representatives by $R_{Q'}$. These are the only hyperplanes that are relevant to the binary search of Lemma 9.2, and therefore one of them will provide the final witness in the binary search (the point r in the proof of Lemma 9.2). This implies that $R_{Q'}$ constitutes a valid representative set for

ε-approximate nearest neighbor searching for any query point that lies in Q'. Thus, the resulting data structure is a valid AVD structure.

In order to bound the query time we recall some of the observations made in the proof of Lemma 9.2. Since K_Q is contained within a hypercube of constant diameter centered at the origin, the absolute slopes of the hyperplanes of the approximating polytope are bounded above by some constant. Recall that the leaf cells of T_Q that contribute a point to $R_{Q'}$ have side lengths at least as large as that of Q'. By the same reasoning used in Lemma 3 of [11], the number of such quadtree leaf cells that can intersect ∂K_Q is bounded by a constant, which we denote by c_ℓ. (This constant depends on the dimension d and the largest possible slope.) Therefore, the total number of cells contributing a representative to $R_{Q'}$ is at most c_ℓ. Since each cell contributes at most $t_{d+1}(\varepsilon)$ representatives, the total number of representatives associated with any leaf cell of T'_Q is at most $c_\ell \cdot t_{d+1}(\varepsilon) = O(t_{d+1}(\varepsilon))$.

The bound on the total space is complicated by the fact that a large cell that intersects ∂K_Q may overlap the columns of many small leaf cells, and hence a large cell's representatives may be replicated many times. Let M denote the set of internal nodes of T_Q all of whose children are leaves. We encountered this set earlier in the proof of Lemma 3.1. As we saw in that earlier lemma, because each node of M was split by SplitReduce, it follows that each such cell requires more than $t_{d+1}(\varepsilon)$ halfspaces to approximate $K(Q)$, and thus, the children of M together require at least as many representatives. Therefore we have $|M| \cdot t_{d+1}(\varepsilon) \leq s_{d+1}(\varepsilon)$. Reasoning as we did in Lemma 3.1, every internal node of T_Q either is in M or is an ancestor of a node in M. Thus, the number of internal nodes is at most $|M| \cdot \text{height}(T_Q)$. Since every internal node has 2^d children, the total number of nodes in T_Q is at most $2^d \cdot |M| \cdot \text{height}(T_Q)$. Clearly, the number of leaf cells of T'_Q can be no larger. As we saw in the previous paragraph, each leaf cell of T'_Q is associated with at most $c_\ell \cdot t_{d+1}(\varepsilon)$ representatives. Since the tree is of height $O(\log \frac{1}{\varepsilon})$, the total number of representatives over all these cells is at most

$$\begin{aligned}(2^d \cdot |M| \cdot \text{height}(T_Q))(c_\ell \cdot t_{d+1}(\varepsilon)) &= c_\ell \cdot 2^d \cdot \text{height}(T_Q) \cdot (|M| \cdot t_{d+1}(\varepsilon)) \\ &\leq \left(c_\ell \cdot 2^d \cdot \log \frac{1}{\varepsilon}\right) \cdot s_{d+1}(\varepsilon) \\ &= O\left(s_{d+1}(\varepsilon) \cdot \log \frac{1}{\varepsilon}\right).\end{aligned}$$

By Lemma 9.1(iii), the total number of representatives in T_Q is $O(n \log \frac{1}{\varepsilon})$. By a counting argument, the number of leaf cells with more than $t_{d+1}(\varepsilon) \cdot \log \frac{1}{\varepsilon}$ representatives is $O(n/t_{d+1}(\varepsilon))$. Therefore, the total space is

$$O\left(n \log \frac{1}{\varepsilon} + \frac{n}{t_{d+1}(\varepsilon)} \cdot s_{d+1}(\varepsilon) \cdot \log \frac{1}{\varepsilon}\right) \;=\; O\left(n \left(1 + \frac{s_{d+1}(\varepsilon)}{t_{d+1}(\varepsilon)}\right) \log \frac{1}{\varepsilon}\right)$$

as desired. □

By combining this with Theorem 1.2 (applying the more accurate space bounds from Lemma 6.4) we obtain the main result of this section.

LEMMA 9.5. *Let $0 < \varepsilon \leq 1$ be a real parameter, $\alpha \geq 1$ be a real constant, and X be a set of n points in $\mathbb{R}^d$. There is a data structure in the AVD model for approximate nearest neighbor searching that achieves*

$$\textit{Query time: } O\left(\log n + (1/\varepsilon^{d/2\alpha}) \cdot \log^2 \frac{1}{\varepsilon}\right),$$

$$\textit{Space: } O\left(n \cdot \max\left(\log \frac{1}{\varepsilon}, 1/\varepsilon^{d\left(\frac{1}{2} - \frac{1}{2\alpha}\right)}\right)\right) \quad \textit{for } 1 \le \alpha < 2, \textit{ and}$$

$$O\left(n/\varepsilon^{d\left(1 - \frac{\lfloor \lg \alpha \rfloor}{\alpha} - \frac{1}{2^{\lfloor \lg \alpha \rfloor}} + \frac{1}{2\alpha}\right)}\right) \quad \textit{for } \alpha \ge 2.$$

The constant factors in the space and query time depend only on d *and* α *(not on* ε*). At the expense of increasing the query time and space by a factor of* $O(\log \frac{1}{\varepsilon})$ *it is possible to construct the data structure in time* $O(n(\log n + 1/\varepsilon^{c\,d}) \log \frac{1}{\varepsilon})$ *for some constant* c *(that does not depend on* d *or* α*).*

Proof. Given X and ε, we first observe that if $1/16 < \varepsilon \le 1$, we may set $\varepsilon = 1/16$, since this will only affect the constant factors in the asymptotic bounds. We consider two cases based on the value of α.

If $1 \le \alpha < 2$, we will apply Theorem 1.2 with the values of d and α of the theorem set to $d' = d + 1$ and $\alpha' = 4$, respectively. The theorem states that there is a data structure that achieves query time

$$O\left(\left(\log \frac{1}{\varepsilon}\right) / \varepsilon^{\frac{d'-1}{\alpha'}}\right) \;=\; O\left((1/\varepsilon^{d/4}) \cdot \log \frac{1}{\varepsilon}\right) \;=\; O\left((1/\varepsilon^{d/2\alpha}) \cdot \log \frac{1}{\varepsilon}\right) \tag{9}$$

and space

$$O\left(1/\varepsilon^{(d'-1)\left(1 - \frac{2\lfloor \lg \alpha' \rfloor - 2}{\alpha'}\right)}\right) \;=\; O\big(1/\varepsilon^{d/2}\big). \tag{10}$$

Letting $t_{d+1}(\varepsilon)$ and $s_{d+1}(\varepsilon)$ denote the quantities of (9) and (10), respectively, we apply Lemma 9.4 to obtain a data structure in the AVD model with query time $O(\log n + (1/\varepsilon^{d/2\alpha}) \cdot \log^2 \frac{1}{\varepsilon})$ and space

$$O\left(n\left(1 + \frac{1/\varepsilon^{d/2}}{(1/\varepsilon^{d/2\alpha}) \cdot \log \frac{1}{\varepsilon}}\right) \log \frac{1}{\varepsilon}\right) \;=\; O\left(n \cdot \max\left(\log \frac{1}{\varepsilon}, 1/\varepsilon^{d\left(\frac{1}{2} - \frac{1}{2\alpha}\right)}\right)\right),$$

as desired.

Otherwise, if $\alpha \ge 2$, we apply Theorem 1.2 (but using the more accurate space bounds from Lemma 6.4) in dimension $d' = d + 1$ and with trade-off parameter $\alpha' = 2\alpha$. (Observe that $\alpha' \ge 4$, as required by Theorem 1.2 and Lemma 6.4.) This yields an approximate polytope membership data structure with query time $t_{d+1}(\varepsilon) = O((1/\varepsilon^{d/2\alpha}) \cdot \log \frac{1}{\varepsilon})$ and space

$$s_{d+1}(\varepsilon) \;=\; O\left(1/\varepsilon^{d\left(1 - 2\left(\frac{\lfloor \lg(2\alpha) \rfloor - 2}{2\alpha} + \frac{1}{2^{\lfloor \lg(2\alpha) \rfloor}}\right)\right)}\right) \;=\; O\left(1/\varepsilon^{d\left(1 - \frac{\lfloor \lg \alpha \rfloor - 1}{\alpha} - \frac{1}{2^{\lfloor \lg \alpha \rfloor}}\right)}\right).$$

By Lemma 9.4 this implies the existence of a data structure in the AVD model with the desired query time of $O(\log n + (1/\varepsilon^{d/2\alpha}) \cdot \log^2 \frac{1}{\varepsilon})$ and space

$$O\left(n\left(1 + \frac{1/\varepsilon^{d\left(1 - \frac{\lfloor \lg \alpha \rfloor - 1}{\alpha} - \frac{1}{2^{\lfloor \lg \alpha \rfloor}}\right)}}{(\log \frac{1}{\varepsilon})/\varepsilon^{d/2\alpha}}\right) \log \frac{1}{\varepsilon}\right).$$

Since $\alpha \geq 2$, we may ignore the "1+" term in the inner parenthetical factor. After some simplification we obtain the desired space bound of

$$O\left(n/\varepsilon^{d\left(1-\frac{\lfloor \lg \alpha \rfloor}{\alpha}-\frac{1}{2^{\lfloor \lg \alpha \rfloor}}+\frac{1}{2\alpha}\right)}\right).$$

The preprocessing involves first computing the AVD, which by Lemma 9.1 takes $O(n \cdot \log n \cdot \log \frac{1}{\varepsilon})$ time. For each of the $O(n \log \frac{1}{\varepsilon})$ leaf cells Q of the AVD, we apply SplitReduce in dimension $d+1$ to its associated set R_Q of representatives. By Lemma 7.6 this takes $O(n_Q + 1/\varepsilon^{c_p(d+1)})$ time, where $n_Q = |R_Q|$, and c_p is a constant that does not depend on d. Summing over all the leaf cells of the AVD and recalling that the total number of representatives is $O(n \cdot \log \frac{1}{\varepsilon})$, it follows that the total preprocessing time is on the order of

$$\begin{aligned} n \cdot \log n \cdot \log \frac{1}{\varepsilon} \;+\; \sum_{Q} \left(n_Q + 1/\varepsilon^{c_p(d+1)}\right) &= n \cdot \log n \cdot \log \frac{1}{\varepsilon} + n \cdot \log \frac{1}{\varepsilon} \cdot \left(\frac{1}{\varepsilon}\right)^{c_p(d+1)} \\ &= n\left(\log n + \left(\frac{1}{\varepsilon}\right)^{cd}\right) \log \frac{1}{\varepsilon}, \end{aligned}$$

where $c = c_p(d+1)/d$, as desired. Because of the reliance on approximate set cover in the processing of Lemma 7.6, the query time and space are larger by a factor of $O(\log \frac{1}{\varepsilon})$. ∎

Note that the above proof uses the AVD-based reduction given in Lemma 9.4. If instead we had used Lemma 9.3, we would obtain a slight improvement in the space, by a factor of $\Theta(\log \frac{1}{\varepsilon})$, at the loss of having a data structure in the AVD model. By the simple observation that $1/2^{\lfloor \lg \alpha \rfloor} \geq 1/\alpha$, the above space bound for the $\alpha \geq 2$ case simplifies to $O(n/\varepsilon^{d(1-\frac{\lfloor \lg \alpha \rfloor}{\alpha}-\frac{1}{2\alpha})})$, and this establishes Theorem 1.4.

10. Proof of the area-product bound. In this section, we present lower bounds for the product of the area of (restricted) ε-dual caps and the associated Voronoi patches, and in particular, we present a proof of Lemma 5.2, which appeared at the end of section 5.

We begin by recalling some notation. We are given a convex body K in $\mathbb{R}^d$ and a pair $(p, h(p))$, where $p \in \partial K$ and $h(p)$ is a supporting hyperplane passing through p, such that p lies within a unit ball centered at the origin. Also recall that p_ε denotes the point lying at distance ε from p in the direction of the outward normal orthogonal to $h(p)$ at p. S denotes the Dudley hypersphere, which is centered at the origin and is of radius 3. For $y \geq 1$, let $H^{(y)}(p)$ be any hyperplane that is parallel to $h(p)$ and translated away from K by distance y. (This is illustrated in Figure 18. Note that the figures of this section are not drawn to scale.) To simplify our descriptions, we consider the directed line segment from p to p_ε to be "vertically downward," so that the hyperplanes $h(p)$ and $H^{(y)}(p)$ are "horizontal" with $h(p)$ above $H^{(y)}(p)$.

Recall that the *ε-dual cap* defined by p, denoted $D(p)$, is the portion of ∂K that is visible from p_ε (see Figure 19(a)). Also, recall that $\mathrm{Vor}(D(p))$ consists of the points that are exterior to K whose closest point on ∂K lies within $D(p)$. Define the *base* of $D(p)$, denoted $\Gamma(p)$, to be the intersection of $h(p)$ with the convex hull of $K \cup \{p_\varepsilon\}$.

For $\delta > 0$, recall that the *δ-restricted ε-dual cap* defined by p, denoted $D_\delta(p)$, is $D(p) \cap B_\delta(p)$, where $B_\delta(p)$ is the Euclidean ball of radius δ centered at p

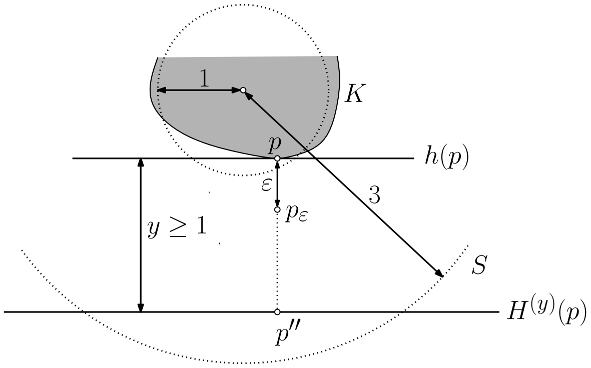


FIG. 18. *Definitions of $h(p)$, $H^{(y)}(p)$, and S.*

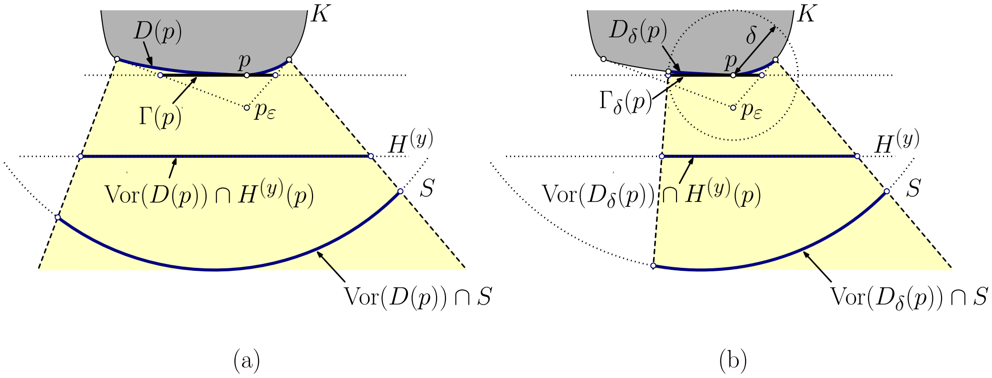

D(p)
K
p
Γ(p)
p_ε
$H^{(y)}$
$\mathrm{Vor}(D(p)) \cap H^{(y)}(p)$
S
$\mathrm{Vor}(D(p)) \cap S$
(a)
$D_\delta(p)$
δ
K
p
$\Gamma_\delta(p)$
p_ε
$H^{(y)}$
$\mathrm{Vor}(D_\delta(p)) \cap H^{(y)}(p)$
S
$\mathrm{Vor}(D_\delta(p)) \cap S$
(b)

FIG. 19. *Dual caps, bases, and Voronoi regions for the* (a) *unrestricted and* (b) *restricted cases.*

(see Figure 19(b)). As before, $\mathrm{Vor}(D_\delta(p))$ is the set of points that are exterior to K whose closest point on ∂K lies within $D_\delta(p)$. Also, the *δ-restricted base*, denoted $\Gamma_\delta(p)$ is $\Gamma(p) \cap B_\delta(p)$.

Our objective in this section is to establish bounds on the product of the area of a $\sqrt{\varepsilon}$-restricted ε-dual cap and its Voronoi patch on the Dudley hypersphere. It will be easier to start with hyperplane patches on $H^{(y)}(p)$ and then generalize to spherical patches on S. The main result of this section is given in the following lemma. Part (ii) is equivalent to Lemma 5.2, which is our main objective. Part (i) is a useful intermediate result.

LEMMA 10.1. *Let K be a convex body in $\mathbb{R}^d$, and let $0 < \varepsilon \leq 1/8$ and $\delta = \sqrt{\varepsilon}$. There are constants c_a and c'_a (depending only on d) such that for any point $p \in \partial K$,*

(i) *given any $y \geq 1$, $\mathrm{area}(D_\delta(p)) \cdot \mathrm{area}(\mathrm{Vor}(D_\delta(p)) \cap H^{(y)}(p)) \geq c'_a \cdot \varepsilon^{d-1}$;*

(ii) *if K is fat and has diameter at least 2ε, and p lies within a unit ball centered at the origin, then $\mathrm{area}(D_\delta(p)) \cdot \mathrm{area}(\mathrm{Vor}(D_\delta(p)) \cap S) \geq c_a \cdot \varepsilon^{d-1}$.*

This lemma holds generally for any $\delta \geq \sqrt{\varepsilon}$, but it suffices for our purposes to consider the restricted case of $\delta = \sqrt{\varepsilon}$. Note that the additional assumptions on fatness and diameter of part (ii) are necessary for establishing a lower bound. If K is not fat or not of sufficiently large diameter, then $\mathrm{area}(D_\delta(p))$ can be arbitrarily small. Since the Dudley hypersphere is bounded, it would not be possible to establish any lower bound on the product of their areas.

The remainder of this section is devoted to proving this lemma. Because p will be fixed throughout, in order to simplify the notation, we will drop references to p. For example, we will use h, $H^{(y)}$, D_δ, Γ_δ, and B_δ in place of $h(p)$, $H^{(y)}(p)$, $D_\delta(p)$, $\Gamma_\delta(p)$, and $B_\delta(p)$, respectively.

Since it will be useful to relate sets on h with sets on $H^{(y)}$, we observe that each of these hyperplanes can be consistently identified with $\mathbb{R}^{d-1}$ by endowing them with parallel coordinate frames, one centered at p (for h) and one centered at p's orthogonal projection onto $H^{(y)}$. Thus, a point on h and its vertical projection onto $H^{(y)}$ have the same coordinates.

We start by proving Lemma 10.1(i). Since the value of y will be fixed throughout this part of the proof, we refer to $H^{(y)}$ simply as H. Let p'' denote the origin of H's coordinate system (the vertical projection of p onto H). (See Figure 18.) In order to exploit Lemma 2.2 on the Mahler volume, rather than considering $\mathrm{Vor}(D_\delta) \cap H$ directly, we will find it convenient to instead analyze the polar dual of the base Γ_δ. Using the aforementioned coordinate frame, we can think of Γ_δ as a body in $\mathbb{R}^{d-1}$. For $r = \sqrt{\varepsilon/8}$, consider the generalized polar of the dual base, $\mathrm{polar}_r(\Gamma_\delta)$, which we can think of as a convex subset of H. Because Γ_δ contains the origin of h (namely, p), it follows directly that $\mathrm{polar}_r(\Gamma_\delta)$ is bounded and convex and also contains the origin of H (namely, p''). In order to obtain a lower bound on $\mathrm{area}(\mathrm{Vor}(D_\delta) \cap H)$, we will first show that $\mathrm{polar}_r(\Gamma_\delta)$ is a subset of $\mathrm{Vor}(D_\delta) \cap H$ and then derive a lower bound on $\mathrm{area}(\mathrm{polar}_r(\Gamma_\delta))$. The first assertion is established by the following lemma.

LEMMA 10.2. *Given the preconditions of Lemma* 10.1 *and* $r = \sqrt{\varepsilon/8}$*, we have* $\mathrm{polar}_r(\Gamma_\delta) \subseteq \mathrm{Vor}(D_\delta) \cap H$.

The proof is rather technical and involves a reduction to the problem in two-dimensional space. Before giving the proof, it will help to provide some intuition regarding the relationship between $\mathrm{Vor}(D_\delta) \cap H$ and the polar of Γ_δ.

For the sake of simplicity, let us consider just the two-dimensional setting. Let t denote a point of tangency on ∂K with respect to p_ε (see Figure 20), and let v be the intersection of the line segment $p_\varepsilon t$ with h. Shoot a ray from t perpendicular to ∂K until it intersects H. Let q denote this intersection point. Since K is convex, all the points on the segment $p''q$ have their nearest neighbor on the portion of ∂K between p and t, that is, they all lie within $\mathrm{Vor}(D)$. Observe that if we translate this perpendicular line so that it emanates from p_ε instead of t, it will hit H at a point q' that is closer to p''. Therefore, the segment $p''q'$ also lies within $\mathrm{Vor}(D)$. Let ℓ denote the distance between p_ε and p''. By similar triangles, it is easy to see that the length of $p''q'$ is $\ell \cdot \varepsilon/\|pv\|$. Since $v \in \Gamma$, q' lies within $\mathrm{polar}_{r'}(\Gamma)$, where $r' = \sqrt{\ell \cdot \varepsilon}$.

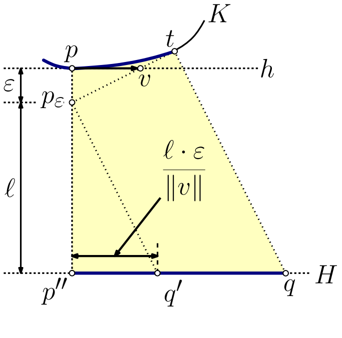
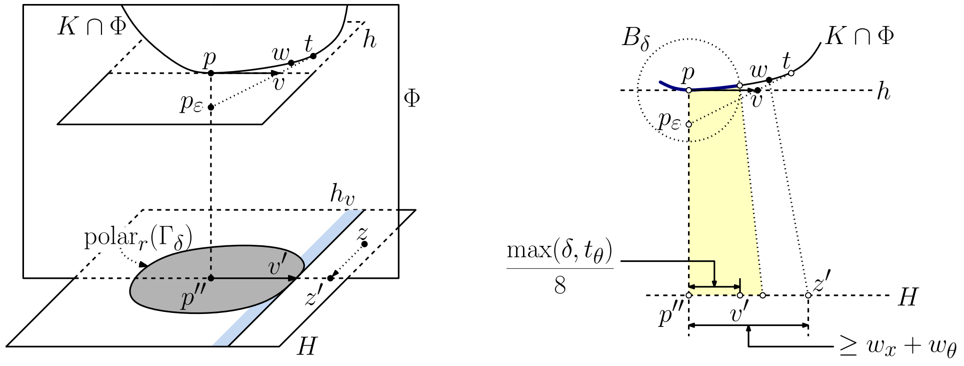


FIG. 20. *The relationship between* $\mathrm{Vor}(D_\delta) \cap H$ *and the polar of* Γ_δ.



(a) (b)

FIG. 21. *The reduction to the plane* Φ.

Because $y \geq 1$ and $\varepsilon \leq 1/8$, we have $r' = \Omega(\sqrt{\varepsilon})$. This observation generalizes readily to higher dimensions, and it follows that $\mathrm{polar}_{r'}(\Gamma) \subseteq \mathrm{Vor}(D) \cap H$. We will show how to generalize this intuition to higher dimensions and the δ-restricted setting.

For any $z \in H$ let w denote its nearest neighbor on ∂K. In order to prove Lemma 10.2, it suffices to show that if $w \notin D_\delta$ (implying that $z \notin \mathrm{Vor}(D_\delta) \cap H$), then $z \notin \mathrm{polar}_r(\Gamma_\delta)$. By our assumption that H lies below K it follows that w lies on the "lower surface" of ∂K (meaning that a vertical ray directed downward from w does not intersect the interior of K). Since w is not in the restricted cap, we know that either $w \notin D$ or $w \notin B_\delta$.

It will simplify the analysis to reduce the problem to a two-dimensional setting. Consider the plane Φ that contains the points p, p_ε, and w. (Note that these points are not collinear.) Let t be the point of tangency on $\partial K \cap \Phi$ with respect to p_ε that lies on the same side as w (see Figure 21(a)). Let v be the intersection of the line segment $p_\varepsilon t$ with h. We may identify Φ with $\mathbb{R}^2$ by imposing a coordinate system on Φ where the origin is at p, the y-axis is directed upward away from p_ε, and the x-axis is parallel to the vector from p to v. Given a point $u \in \Phi$, let u_x and u_y denote its coordinates relative to this coordinate system. Further, if $u \in \partial K \cap \Phi$, let u_θ denote the slope of the (unique) supporting line on Φ passing through u. Note that z need not lie on Φ. Let z' be the orthogonal projection of z onto Φ. Observe that $t_\theta = \varepsilon/\|pv\|$, and therefore $\|pv\| = \varepsilon/t_\theta$. By our choice of coordinate system and assumptions about orientations, the coordinates of w, t and the slopes w_θ and t_θ are all nonnegative quantities.

The point v lies on the base Γ of p's unrestricted dual cap. By employing our coordinate system on h, we can identify v with a vector in $\mathbb{R}^{d-1}$ (emanating from p). If $\|pv\| \leq \delta$, then v contributes a bounding halfspace to $\mathrm{polar}_r(\Gamma_\delta)$. This halfspace is bounded by a hyperplane that is orthogonal to v and lies at distance $r^2/\|pv\|$ from the origin. Let us think of this halfspace, which we denote by h_v, as lying on H (see Figure 21(a)). Recalling that $r = \sqrt{\varepsilon/8}$, the distance of h_v's bounding hyperplane to the origin p'' is $(\varepsilon/8)/\|pv\| = t_\theta/8$. On the other hand, if $\|pv\| > \delta$, then v lies outside the restricted base. In this case v's subvector of length δ lies on the boundary of the restricted base and contributes to $\mathrm{polar}_r(\Gamma_\delta)$ a halfspace whose bounding hyperplane is at distance $(\varepsilon/8)/\delta$ from the origin. Recalling that $\delta = \sqrt{\varepsilon}$, this is equal to $\delta/8$. Thus, in either case, $\mathrm{polar}_r(\Gamma_\delta)$ is bounded by a halfspace whose defining hyperplane is orthogonal to v and lies at distance $\max(\delta, t_\theta)/8$ from the origin. This hyperplane intersects the horizontal line $H \cap \Phi$ at some point v' that lies to the right of p'' at distance $v'_x = \max(\delta, t_\theta)/8$ (see Figure 21(b)).

Because the hyperplane passing through v' is orthogonal to v, in order to show that $z \notin \mathrm{polar}_r(\Gamma_\delta)$, it suffices to show that z' does not lie within h_v, which is equivalent to showing that $z'_x > v'_x$. We have thus reduced the problem to a two-dimensional setting.

Recall that w is the closest point to z on ∂K. We assert that w is also the closest point to z' on $\partial K \cap \Phi$. The reason is that the squared distance from z to any point on $\partial K \cap \Phi$ can be expressed as the sum of the squared distance from z' to this point and the squared distance from z to z'. Since the latter quantity is the same for all points on Φ, the closest point to z is also the closest point to z'. From basic properties of convexity, it follows that the line wz' is orthogonal to the support line passing through w on $\partial K \cap \Phi$. Therefore, the slope of wz' (in Φ's coordinate system) is $-1/w_\theta$, and in particular we have $(z'_x - w_x)/(z'_y - w_y) = -w_\theta$. Since h and H are separated by at least unit distance (with h above H), we have $w_y - z'_y \geq 1$, and so $z'_x \geq w_x + w_\theta$.

Thus, to complete the proof of Lemma 10.2, it suffices to show that if $w \notin D_\delta$, then $v'_x < w_x + w_\theta$. We first establish two useful technical results. These results will be applied in a context where w lies within the unrestricted dual cap but outside the restricted dual cap, that is, when $w_x \leq t_x$ but $w \notin B_\delta$. The first result shows that if t_θ is sufficiently small, the slope of the line pw is at most unity. The second shows that if t_θ is sufficiently large, the slope of pw is not much smaller than the slope of t's supporting line.

LEMMA 10.3. *Given the preconditions of Lemma* 10.1 *and the aforementioned two-dimensional reduction, and given w and t as introduced above, where $w_x \leq t_x$ and $w \notin B_\delta$,*
(i) *if $t_\theta \leq \delta\sqrt{8}$, then $w_y/w_x \leq 1$, and*
(ii) *if $t_\theta > \delta\sqrt{8}$, then $w_y/w_x \geq t_\theta/2$.*

The proof is a straightforward geometric exercise and has been omitted. (See [8] for the full proof.)

We are now in position to complete the proof of Lemma 10.2. Recall that our objective is to show that if $w \notin D_\delta$, then $v'_x < w_x + w_\theta$, where $v'_x = \max(\delta, t_\theta)/8$. We consider two cases, depending on t_θ. First, if $t_\theta \leq \delta\sqrt{8}$, then $v'_x \leq \max(\delta, \delta\sqrt{8})/8 = \delta/\sqrt{8}$. Since the line $p_\varepsilon t$ has slope t_θ and $t_y \geq 0$, we have $t_x = (t_y + \varepsilon)/t_\theta \geq \varepsilon/t_\theta \geq \delta/\sqrt{8}$. We consider two subcases. If $w_x > t_x$, then we have $w_x + w_\theta > t_x \geq \delta/\sqrt{8} \geq v'_x$, as desired. On the other hand, if $w_x \leq t_x$, then w is inside the unrestricted cap D. Since by our hypothesis, w is not in the restricted cap, it must be that $w \notin B_\delta$, that is, $w_x^2 + w_y^2 > \delta^2$. By Lemma 10.3(i), we have $w_x \geq w_y$. Therefore, $2w_x^2 \geq w_x^2 + w_y^2 > \delta^2$, which implies that $w_x > \delta/\sqrt{2}$. Therefore, $w_x + w_\theta \geq w_x > \delta/\sqrt{2} > v'_x$, as desired.

For the second case, assume that $t_\theta > \delta\sqrt{8}$. In this case $v'_x = t_\theta/8$. As before, we consider two subcases. If $w_x > t_x$, then by convexity $w_\theta \geq t_\theta$, and so $w_x + w_\theta \geq t_\theta > v'_x$, as desired. On the other hand, if $w_x \leq t_x$, then since w lies within the unrestricted cap, we may infer that $w \notin B_\delta$. By Lemma 10.3(ii), we have $w_y/w_x \geq t_\theta/2$. Because the support line at w passes below the origin, we also have $w_\theta \geq w_y/w_x$. Therefore $w_x + w_\theta \geq w_y/w_x \geq t_\theta/2 > v'_x$. This completes the proof of Lemma 10.2.

Because it is easier to deal with flat objects than curved ones, before returning to the proof of Lemma 10.1(i), we show that the area of the restricted dual cap is, up to a constant factor, bounded below by the area of its base. This result is straightforward for unrestricted caps, since it is easy to show that the base is contained within the orthogonal projection of the dual cap onto h. However, restriction complicates the analysis. The proof involves a rather technical but straightforward geometric argument. We have omitted it, but it is given in full in [8].

Lemma 10.4. *Given the preconditions of Lemma* 10.1*, it follows that* $\mathrm{area}(D_\delta) \geq \mathrm{area}(\Gamma_\delta)/2^{d-1}$.

We are now ready to prove Lemma 10.1(i). Recall that $r = \sqrt{\varepsilon/8}$. As observed earlier, $\mathrm{polar}_r(\Gamma_\delta)$ is a scaled copy of $\mathrm{polar}(\Gamma_\delta)$ by a factor of r^2, and therefore (since these are $(d-1)$-dimensional bodies) we have $\mathrm{area}(\mathrm{polar}_r(\Gamma_\delta)) = r^{2(d-1)} \cdot \mathrm{area}(\mathrm{polar}(\Gamma_\delta))$. By applying Lemma 10.2, we have

$$\mathrm{area}(\mathrm{Vor}(D_\delta) \cap H) \;\geq\; \mathrm{area}(\mathrm{polar}_r(\Gamma_\delta)) \;=\; r^{2(d-1)} \cdot \mathrm{area}(\mathrm{polar}(\Gamma_\delta)).$$

By Lemma 10.4, $\mathrm{area}(D_\delta) \geq \mathrm{area}(\Gamma_\delta)/2^{d-1}$, and therefore

$$\begin{aligned}\mathrm{area}(D_\delta) \cdot \mathrm{area}(\mathrm{Vor}(D_\delta) \cap H) &\geq \frac{\mathrm{area}(\Gamma_\delta)}{2^{d-1}} \cdot r^{2(d-1)} \cdot \mathrm{area}(\mathrm{polar}(\Gamma_\delta)) \\ &\geq \left(\frac{r^2}{2}\right)^{d-1} \mathrm{area}(\Gamma_\delta) \cdot \mathrm{area}(\mathrm{polar}(\Gamma_\delta)).\end{aligned}$$

We now apply the Mahler-volume bound. By Lemma 2.2 (in $\mathbb{R}^{d-1}$), there exists a constant c_m (depending only on d) such that $\mathrm{area}(\Gamma_\delta) \cdot \mathrm{area}(\mathrm{polar}(\Gamma_\delta)) \geq c_m$. Therefore,

$$\mathrm{area}(D_\delta) \cdot \mathrm{area}(\mathrm{Vor}(D_\delta) \cap H) \;\geq\; c_m \left(\frac{r^2}{2}\right)^{d-1} \;=\; c_m \left(\frac{\varepsilon}{16}\right)^{d-1}.$$

Selecting any $c'_a \leq c_m/16^{d-1}$ establishes Lemma 10.1(i).

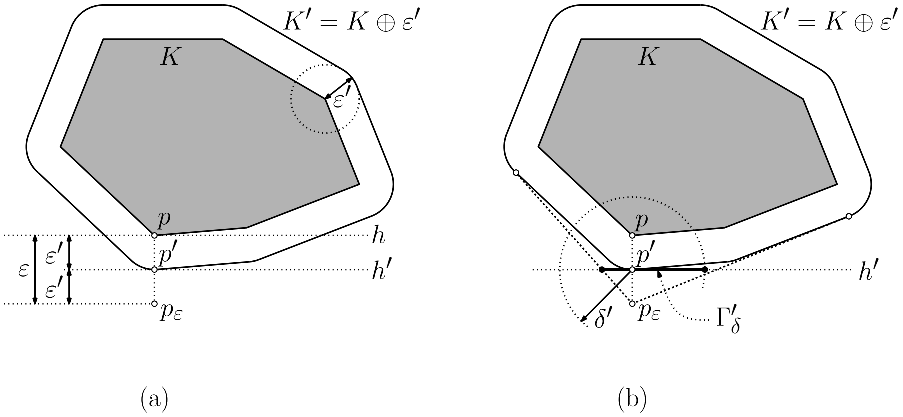
Next, let us establish Lemma 10.1(ii). Recall that we assume that K is fat and of diameter at least 2ε. In particular, let us assume that K is γ-fat, where γ is a constant independent of n and ε that lies in the interval $(0, 1]$. (As a result of Lemma 2.1, we may assume that γ is $1/d$ when applying this result.)

It is natural to try to generalize the approach used in part (i). First, we would show that

$$\mathrm{area}(\mathrm{Vor}(D_\delta) \cap S) = \Omega\big(r^{2(d-1)} \cdot \mathrm{area}(\mathrm{polar}(\Gamma_\delta))\big) \quad \text{and} \quad \mathrm{area}(D_\delta) = \Omega(\mathrm{area}(\Gamma_\delta)),$$

and then we would apply the Mahler-volume bound to yield a lower bound on the product $\mathrm{area}(\Gamma_\delta) \cdot \mathrm{area}(\mathrm{polar}(\Gamma_\delta))$. A problem arises, however, if K is not smooth. In particular, if some portion of the boundary of K in p's vicinity is nearly vertical, then the boundary of Γ_δ can be arbitrarily close to the origin (namely, p), implying that $\mathrm{polar}(\Gamma_\delta)$ cannot be bounded, and hence its area can be arbitrarily large. This was not an issue in part (i), because H is also unbounded. But since S is bounded, $\mathrm{area}(\mathrm{Vor}(D_\delta) \cap S)$ cannot be arbitrarily large. We will remedy this by smoothing K by taking its Minkowski sum with a small Euclidean ball of radius $O(\varepsilon)$. We shall see (in the proof of Lemma 10.7) that this allows us to constrain the area of $\mathrm{polar}(\Gamma_\delta)$. This smoothing operation requires us to adapt many of the prior results of this section to this new context.

To construct the smoothed body, for the remainder of this section define $\varepsilon' = \varepsilon/2$, and let $K' = K \oplus \varepsilon'$ (see Figure 22(a)). Recall that h denotes the supporting hyperplane at p and p_ε is the point at distance ε from p in the direction orthogonal to h. As before, for the sake of illustration, let us assume that p_ε is vertically below p.



(a) (b)

FIG. 22. *The smoothed body K'.*

Let p' be the midpoint of the segment pp_ε. Clearly, $p' \in \partial K'$, and the parallel hyperplane h' passing through p' is a supporting hyperplane for K'.

Let us also define the dual base in this smoothed context. Define Γ' to be the intersection of h' and $\text{conv}(K' \cup \{p_\varepsilon\})$. Let $\delta' = \sqrt{\varepsilon'} = \delta/\sqrt{2}$, and define the restricted base Γ'_δ to be the intersection of Γ' and a ball of radius δ' centered at p' (see Figure 22(b)). Our analysis will be based on K' and Γ'_δ, as opposed to K and Γ_δ. Our first objective will be to show that the area of Γ'_δ is not significantly larger than that of Γ_δ. As before, we endow h and h' with parallel coordinate frames whose origins are located at p and p', respectively. Then we can think of Γ_δ and Γ'_δ as convex sets in $\mathbb{R}^{d-1}$. The following lemma relates these two bodies.

LEMMA 10.5. *Given a convex body K that is γ-fat and of diameter at least 2ε and given Γ_δ and Γ'_δ as defined above, there exists a constant c (depending on γ and the dimension d) such that* $\text{area}(\Gamma'_\delta) \leq c \cdot \text{area}(\Gamma_\delta)$.

This result is not surprising, given that K is fat and $\|pp'\|$ is within a constant factor of $\|pp_\varepsilon\|$. The proof involves straightforward geometric reasoning, but (as always) restriction complicates the analysis. It is presented in full in [8].

Recall that $\text{Vor}(D_\delta) \cap S$ consists of the set of points on the sphere S whose closest point on ∂K lies within the restricted dual cap D_δ. Let D'_δ denote the corresponding restricted dual cap for K', that is, the set of points of $\partial K'$ that are visible from p_ε and lie within the ball $B_{\delta'}(p')$. Our analysis will be based on establishing a lower bound on the area of $\text{Vor}(D'_\delta) \cap S$. The following lemma shows that this will provide a lower bound on the area of $\text{Vor}(D_\delta) \cap S$.

LEMMA 10.6. *Given the preconditions of Lemma* 10.1(ii), $\text{area}(\text{Vor}(D'_\delta) \cap S) \leq \text{area}(\text{Vor}(D_\delta) \cap S)$.

Proof. We sketch the proof here, but complete details can be found in [8]. We prove the stronger result that $\text{Vor}(D'_\delta) \cap S \subseteq \text{Vor}(D_\delta) \cap S$. First observe that (by our restriction on ε and the definition of δ) both D_δ and D'_δ lie within the Dudley hypersphere S. Consider any point $q'' \in \text{Vor}(D'_\delta) \cap S$. It suffices to show that $q'' \in \text{Vor}(D_\delta) \cap S$. Let q and q' be the closest points on ∂K and $\partial K'$, respectively, to q''. It follows from basic properties of the Minkowski sum that these three points are collinear and there are supporting hyperplanes at q and q' that are orthogonal to this line. Since $q'' \in \text{Vor}(D'_\delta) \cap S$, we have $q' \in D'_\delta$, which implies that q' is visible from p_ε. By the existence of parallel supporting hyperplanes, q is also visible from p_ε. Therefore, q lies in p's unrestricted dual cap D. By basic properties of convexity, $\|pq\| \le \|p'q'\|$, which implies that q lies within p's restricted dual cap, and therefore $q'' \in \mathrm{Vor}(D_\delta) \cap S$, as desired. □

Before completing the proof of Lemma 10.1(ii), we exploit the smoothness of K' to establish a relationship between the areas of $\mathrm{Vor}(D'_\delta) \cap S$ and $\mathrm{polar}_{r'}(\Gamma'_\delta)$, where the polar radius r' is suitably modified for the smoothed context. This is given in the next lemma.

LEMMA 10.7. *Given the preconditions of Lemma* 10.1(ii) *and* $r' = \sqrt{\varepsilon'/8}$, *we have* $\mathrm{area}(\mathrm{Vor}(D'_\delta) \cap S) \ge \mathrm{area}(\mathrm{polar}_{r'}(\Gamma'_\delta))$.

Proof. The proof involves a rather technical geometric analysis. We present a sketch here, but complete details can be found in [8]. First, we use the fact that since p is in K, there is a ball of radius $\varepsilon' = \varepsilon/2$ centered at p that lies within K'. It follows that Γ'_δ contains a $(d-1)$-dimensional Euclidean ball (centered at p') of radius $\varepsilon'/\sqrt{3}$.

Let H' denote the hyperplane that is at unit distance below p'. Let p'' denote the vertical projection of p' onto H'. By the definition of the polar transformation, $\mathrm{polar}_{r'}(\Gamma'_\delta)$ (when viewed as a subset of H') is contained within a $(d-1)$-dimensional unit ball centered at p''. Let C denote the semi-infinite generalized cylinder whose horizontal cross section is $\mathrm{polar}_{r'}(\Gamma'_\delta)$, whose upper surface lies on H', and which extends vertically downward. Lemma 10.2 (applied now to K', ε', $\mathrm{polar}_{r'}(\Gamma'_\delta)$ and $\mathrm{Vor}(D'_\delta) \cap H'$) implies that $\mathrm{polar}_{r'}(\Gamma'_\delta) \subseteq \mathrm{Vor}(D'_\delta) \cap H'$. Since this applies not only to H' but to any hyperplane lying below H', it follows that $C \subseteq \mathrm{Vor}(D'_\delta)$. The remainder of the proof involves showing that S is large enough that the orthogonal projection of $S \cap C$ onto H' is equal to $\mathrm{polar}_{r'}(\Gamma'_\delta)$. Since $S \cap C \subseteq \mathrm{Vor}(D'_\delta) \cap S$, and since the area of the orthogonal projection of a set cannot be larger than the area of the original set, we have

$$\mathrm{area}(\mathrm{Vor}(D'_\delta) \cap S) \;\ge\; \mathrm{area}(S \cap C) \;\ge\; \mathrm{area}(\mathrm{polar}_{r'}(\Gamma'_\delta)),$$

as desired. □

We are now ready to prove Lemma 10.1(ii). Recall that $r' = \sqrt{\varepsilon'/8}$. By Lemmas 10.6 and 10.7, we have

$$\mathrm{area}(\mathrm{Vor}(D_\delta) \cap S) \;\ge\; \mathrm{area}(\mathrm{Vor}(D'_\delta) \cap S) \;\ge\; \mathrm{area}(\mathrm{polar}_{r'}(\Gamma'_\delta)).$$

As observed earlier, $\mathrm{polar}_{r'}(\Gamma'_\delta)$ is a scaled copy of $\mathrm{polar}(\Gamma'_\delta)$ by a factor of $(r')^2 = \varepsilon'/8 = \varepsilon/16$, and therefore (since these are $(d-1)$-dimensional bodies) we have

$$\mathrm{area}(\mathrm{Vor}(D_\delta) \cap S) \;\ge\; \left(\frac{\varepsilon}{16}\right)^{d-1} \cdot \mathrm{area}(\mathrm{polar}(\Gamma'_\delta)).$$

By Lemma 10.4, $\mathrm{area}(D_\delta) \ge \mathrm{area}(\Gamma_\delta)/2^{d-1}$. Also, by Lemma 10.5 there is a constant c'' (depending on the fatness parameter γ and d) such that $\mathrm{area}(\Gamma'_\delta) \le c'' \cdot \mathrm{area}(\Gamma_\delta)$. Therefore, we have

$$\mathrm{area}(D_\delta) \;\ge\; \frac{\mathrm{area}(\Gamma_\delta)}{2^{d-1}} \;\ge\; \frac{\mathrm{area}(\Gamma'_\delta)}{c'' \cdot 2^{d-1}}.$$

Combining these, we obtain

$$\begin{aligned}\mathrm{area}(D_\delta) \cdot \mathrm{area}(\mathrm{Vor}(D_\delta) \cap S) &\ge \frac{\mathrm{area}(\Gamma'_\delta)}{c'' \cdot 2^{d-1}} \cdot \left(\frac{\varepsilon}{16}\right)^{d-1} \cdot \mathrm{area}(\mathrm{polar}(\Gamma'_\delta))\\ &= \frac{1}{c''}\left(\frac{\varepsilon}{32}\right)^{d-1} \mathrm{area}(\Gamma'_\delta) \cdot \mathrm{area}(\mathrm{polar}(\Gamma'_\delta)).\end{aligned}$$

By applying Lemma 2.2 (in $\mathbb{R}^{d-1}$) to Γ'_δ, there exists a constant c_m (depending on d) such that

$$\mathrm{area}(\Gamma'_\delta) \cdot \mathrm{area}(\mathrm{polar}(\Gamma'_\delta)) \;\geq\; c_m.$$

Therefore, we have

$$\mathrm{area}(D_\delta) \cdot \mathrm{area}(\mathrm{Vor}(D_\delta) \cap S) \;\geq\; \frac{c_m}{c''} \left(\frac{\varepsilon}{32}\right)^{d-1}.$$

Selecting any $c_a \leq (c_m/c'')(1/32)^{d-1}$ establishes Lemma 10.1(ii). This concludes our proof of the area bounds.

11. Concluding remarks. In this paper we have presented an efficient data structure for determining approximately whether a given query point lies within a convex body. Our solution is based on a simple and natural quadtree-based algorithm, called SplitReduce. Our principal technical contribution has been an analysis of the space-time trade-offs for this algorithm. These are the first nontrivial space-time trade-offs for this problem. We do not know whether this analysis is tight, but we presented a lower bound example that demonstrates the limits of possible improvements. We also demonstrated the value of approximate polytope membership by showing that our data structure can be combined with an AVD data structure to produce significant improvements to the space-time trade-offs of approximate nearest neighbor searching in Euclidean space.

Our analysis of the trade-offs involved a combination of a number of novel techniques, which may be of broader interest. One notable example is the application of the Mahler volume as a means of analyzing the local structure of a convex body through consideration of both its primal and dual representations. This resulted in an efficient two-pronged sampling strategy for computing hitting sets of low cardinality for ε-dual caps. The Mahler volume has also been applied in [7] to derive an optimal area-sensitive bound on the number of facets needed to approximate a convex body.

This work provokes a number of questions for further research. The first involves extending approximate polytope membership queries to other approximate query problems involving convex bodies. For example, in section 9 we showed how to reduce approximate nearest neighbor searching in dimension d to vertical ray shooting queries in dimension $d+1$. However, the polytope involved had a very restricted structure. It would be interesting to know whether there is a data structure exhibiting similar trade-offs for answering approximate ray-shooting queries for general convex bodies. Another example is answering approximate linear-programming queries, where a convex body is preprocessed, and the problem is to determine an extreme point of the body approximately in a given query direction. A further generalization of this would be to extend the work of Barba and Langerman [13] to an approximate setting. It particular, is it possible to preprocess convex bodies so that given two such bodies that have been translated and rotated, it can be decided efficiently whether they intersect each other approximately?

Our result on approximate nearest neighbor searching relies on the lifting transformation to reduce the problem to approximate polytope membership. As a consequence, this approach is applicable only to Euclidean distances. This raises the question of whether there exists a more direct route to approximate nearest neighbor searching that achieves similar space-time improvements and yet avoids reliance on lifting. For example, Arya and Chan [5] have presented improvements to approximate nearest neighbor searching that do not involve lifting. This raises the hope that generalizations to other norms may be possible. While their focus was different from ours (for example, space-time trade-offs are not considered), their results are inferior to our best bounds. These better bounds arise explicitly from concepts like the Mahler volume, which are applicable only in the context of convex approximation, and hence they rely crucially on lifting. A major challenge is whether it is possible to bypass this intermediate step in order to obtain analogous improvements for approximate nearest neighbor searching.

Note added in proof. After the original submission of this paper, the authors discovered a new approach to polytope membership that achieves query time $O(\log \frac{1}{\varepsilon})$ with storage of only $O(1/\varepsilon^{(d-1)/2})$ [9]. As a consequence, it is possible to answer ε-approximate nearest neighbor queries for a set of n points in $O(\log \frac{n}{\varepsilon})$ time with storage of only $O(n/\varepsilon^{d/2})$. While these new results surpass the results of this paper theoretically, the data structure presented there involves significantly larger constant factors and lacks the simplicity and practicality of the approach described here.

Acknowledgment. The authors would like to thank the anonymous reviewers for their many insightful comments.

REFERENCES

[1] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan, *Approximating extent measures of points*, J. Assoc. Comput. Mach., 51 (2004), pp. 606–635.

[2] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan, *Geometric approximation via coresets*, in Combinatorial and Computational Geometry, J. E. Goodman, J. Pach, and E. Welzl, eds., MSRI Publications, Cambridge, UK, 2005.

[3] P. K. Agarwal and J. Matoušek, *Ray shooting and parametric search*, SIAM J. Comput., 22 (1993), pp. 794–806, https://doi.org/10.1137/0222051.

[4] N. Alon and J. H. Spencer, *The Probabilistic Method*, Wiley-Interscience, New York, 2000.

[5] S. Arya and T. M. Chan, *Better ε-dependencies for offline approximate nearest neighbor search, Euclidean minimum spanning trees, and ε-kernels*, in Proceedings of the 30th Annual Symposium on Computational Geometry, 2014, pp. 416–425.

[6] S. Arya, G. D. da Fonseca, and D. M. Mount, *A unified approach to approximate proximity searching*, in Proceedings of the 18th Annual European Symposium on Algorithms, 2010, pp. 374–385.

[7] S. Arya, G. D. da Fonseca, and D. M. Mount, *Optimal area-sensitive bounds for polytope approximation*, in Proceedings of the 28th Annual Symposium on Computational Geometry, 2012, pp. 363–372, https://doi.acm.org/10.1145/2261250.2261305.

[8] S. Arya, G. D. da Fonseca, and D. M. Mount, *Approximate Polytope Membership Queries*, arXiv:1604.01183 [cs.CG], 2016.

[9] S. Arya, G. D. da Fonseca, and D. M. Mount, *Optimal approximate polytope membership*, in Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms, 2017, pp. 270–288.

[10] S. Arya, T. Malamatos, and D. M. Mount, *Space-time tradeoffs for approximate nearest neighbor searching*, J. Assoc. Comput. Mach., 57 (2009), pp. 1–54, https://doi.acm.org/10.1145/1613676.1613677.

[11] S. Arya and D. M. Mount, *Approximate range searching*, Comput. Geom. Theory Appl., 17 (2000), pp. 135–163.

[12] K. Ball, *An elementary introduction to modern convex geometry*, in Flavors of Geometry, S. Levy, ed., Math Sci. Res. Inst. Publ. 31, Cambridge University Press, Cambridge, UK, 1997, pp. 1–58.

[13] L. Barba and S. Langerman, *Optimal detection of intersections between convex polyhedra*, in Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms, 2015, pp. 1641–1654.

[14] G. Barequet and S. Har-Peled, *Efficiently approximating the minimum-volume bounding box of a point set in three dimensions*, J. Algorithms, 38 (2001), pp. 91–109.

[15] J. L. Bentley, M. G. Faust, and F. P. Preparata, *Approximation algorithms for convex hulls*, Commun. ACM, 25 (1982), pp. 64–68, https://doi.acm.org/10.1145/358315.358392.

[16] J. Bourgain and V. D. Milman, *New volume ratio properties for convex symmetric bodies*, Invent. Math., 88 (1987), pp. 319–340, https://doi.org/10.1007/BF01388911.

[17] E. M. Bronshteyn and L. D. Ivanov, *The approximation of convex sets by polyhedra*, Siberian Math. J., 16 (1976), pp. 852–853.

[18] E. M. Bronstein, *Approximation of convex sets by polytopes*, J. Math. Sci., 153 (2008), pp. 727–762.

[19] C. J. C. Burges, *A tutorial on support vector machines for pattern recognition*, Data Min. Knowl. Discov., 2 (1998), pp. 121–167, https://dx.doi.org/10.1023/A:1009715923555.

[20] T. M. Chan, *Fixed-dimensional linear programming queries made easy*, in Proceedings of the 12th Annual Symposium on Computational Geometry, 1996, pp. 284–290, https://doi.acm.org/10.1145/237218.237397.

[21] T. M. Chan, *Output-sensitive results on convex hulls, extreme points, and related problems*, Discrete Comput. Geom., 16 (1996), pp. 369–387.

[22] T. M. Chan, *Closest-point problems simplified on the RAM*, in Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, 2002, pp. 472–473.

[23] T. M. Chan, *Faster core-set constructions and data-stream algorithms in fixed dimensions*, Comput. Geom. Theory Appl., 35 (2006), pp. 20–35, https://dx.doi.org/10.1016/j.comgeo.2005.10.002.

[24] T. M. Chan, *Optimal partition trees*, in Proceedings of the 26th Annual Symposium on Computational Geometry, 2010, pp. 1–10, https://doi.acm.org/10.1145/1810959.1810961.

[25] B. Chazelle and D. P. Dobkin, *Intersection of convex objects in two and three dimensions*, J. Assoc. Comput. Mach., 34 (1987), pp. 1–27.

[26] B. Chazelle and J. Matoušek, *On linear-time deterministic algorithms for optimization problems in fixed dimension*, J. Algorithms, 21 (1996), pp. 579–597, https://doi.org/10.1006/jagm.1996.0060.

[27] K. L. Clarkson, *Algorithms for polytope covering and approximation*, in Proceedings of the 3rd International Workshop on Algorithms of Data Structure, 1993, pp. 246–252.

[28] K. L. Clarkson, *An algorithm for approximate closest-point queries*, in Proceedings of the 10th Annual Symposium of Computational Geometry, 1994, pp. 160–164.

[29] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 2001.

[30] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd ed., Springer, New York, 2010.

[31] D. P. Dobkin and D. G. Kirkpatrick, *Fast detection of polyhedral intersection*, Theoret. Comput. Sci., 27 (1983), pp. 241–253.

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

[33] C. A. Duncan, M. T. Goodrich, and S. Kobourov, *Balanced aspect ratio trees: Combining the advantages of k-d trees and octrees*, J. Algorithms, 38 (2001), pp. 303–333.

[34] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Springer, New York, 1987.

[35] H. G. Eggleston, *Convexity*, Cambridge University Press, Cambridge, UK, 1958.

[36] J. Erickson, L. J. Guibas, J. Stolfi, and L. Zhang, *Separation-sensitive collision detection for convex objects*, in Proceedings of the 10th Annual ACM-SIAM Symposium of Discrete Algorithms, 1999, pp. 327–336.

[37] P. M. Gruber, *Asymptotic estimates for best and stepwise approximation of convex bodies* I, Forum Math., 5 (1993), pp. 521–537.

[38] S. Har-Peled, *A replacement for Voronoi diagrams of near linear size*, in Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science, 2001, pp. 94–103.

[39] S. Har-Peled, *Geometric Approximation Algorithms*, AMS, Providence, RI, 2011.

[40] J. K. Boroczky, *Approximation of general smooth convex bodies*, Adv. Math., 153 (2000), pp. 325–341.

[41] G. Kuperberg, *From the Mahler conjecture to Gauss linking integrals*, Geom. Funct. Anal., 18 (2008), pp. 870–892.

[42] J. Matoušek and O. Schwarzkopf, *On ray shooting in convex polytopes*, Discrete Comput. Geom., 10 (1993), pp. 215–232.

[43] J. Matoušek, *Reporting points in halfspaces*, Comput. Geom. Theory Appl., 2 (1992), pp. 169–186.

[44] J. Matoušek, *Linear optimization queries*, J. Algorithms, 14 (1993), pp. 432–448, https://dx.doi.org/10.1006/jagm.1993.1023.

[45] J. S. B. Mitchell and S. Suri, *Separation and approximation of polyhedral objects*, Comput. Geom. Theory Appl., 5 (1995), pp. 95–114.

[46] E. A. Ramos, *Linear programming queries revisited*, in Proceedings of the 16th Annual Symposium on Computational Geometry, 2000, pp. 176–181, https://doi.acm.org/10.1145/336154.336198.

[47] Y. Sabharwal, S. Sen, and N. Sharma, *Nearest neighbors search using point location in balls with applications to approximate Voronoi decompositions*, J. Comput. System Sci., 72 (2006), pp. 955–977.

[48] L. A. Santaló, *An affine invariant for convex bodies of n-dimensional space*, Port. Math., 8 (1949), pp. 155–161 (in Spanish).