Approximating the Simplicial Depth

Peyman Afshani [1*] *Donald R. Sheehy* [2] *Yannik Stein* [3†]

1 **MADALGO, Department of Computer Science, Aarhus University, Denmark**
peyman@madalgo.au.dk

2 **University of Connecticut, USA**
don.r.sheehy@gmail.com

3 **Institut für Informatik, Freie Universität Berlin, Germany**
yannik.stein@fu-berlin.de

Abstract

Let P be a set of n points in d-dimensions. The simplicial depth, $\sigma_P(q)$ of a point q is the number of d-simplices with vertices in P that contain q in their convex hulls. The simplicial depth is a notion of data depth with many applications in robust statistics and computational geometry. Computing the simplicial depth of a point is known to be a challenging problem. The trivial solution requires $O(n^{d+1})$ time whereas it is generally believed that one cannot do better than $O(n^{d-1})$.

In this paper, we consider approximation algorithms for computing the simplicial depth of a point. For $d = 2$, we present a new data structure that can approximate the simplicial depth in polylogarithmic time, using polylogarithmic query time. In 3D, we can approximate the simplicial depth of a given point in near-linear time, which is clearly optimal up to polylogarithmic factors. For higher dimensions, we consider two approximation algorithms with different worst-case scenarios. By combining these approaches, we compute a $(1+\varepsilon)$-approximation of the simplicial depth in time $\tilde{O}(n^{d/2+1})$ ignoring polylogarithmic factor. All of these algorithms are Monte Carlo algorithms. Furthermore, we present a simple strategy to compute the simplicial depth exactly in $O(n^d \log n)$ time, which provides the first improvement over the trivial $O(n^{d+1})$ time algorithm for $d > 4$. Finally, we show that computing the simplicial depth exactly is #P-complete and W[1]-hard if the dimension is part of the input.

*Work supported in part by the Danish National Research Foundation grant DNRF84 through Center for Massive Data Algorithmics (MADALGO).

†Supported by the Deutsche Forschungsgemeinschaft within the research training group "Methods for Discrete Structures" (GRK 1408).

1 Introduction

Let $P \subset \mathbb{R}^d$ be a point set and $q \in \mathbb{R}^d$ be a point. The *simplicial depth* [31] $\sigma_P(q)$ of q with respect to P is the number of subsets $P' \subseteq P$, $|P'| = d{+}1$, that contain q in their convex hull (see also [12] for an alternate definition). This is one of the important definitions of data depth and has generated interest in both robust statistics and computational geometry since its introduction. Designing efficient algorithms to compute (or approximate) the simplicial depth of a point remains an intriguing task in this area.

Other notions of depth include halfspace (a.k.a. Tukey) depth, Oja depth, regression depth and convex hull peeling depth [5]. Among them, the one most relevant to our techniques is the Tukey depth: given a set P of n points in $\mathbb{R}^d$, the Tukey depth, $\tau_P(q)$ of q with respect to P is the minium number of points contained in a halfspace that also contains q.

Previous and Related Results. Computing the simplicial depth of a single point in 2D was considered even before its formal definition [26] almost three decades ago, perhaps because it translates into an "intuitive" problem of counting the number of triangles containing a given point. In fact, at least three independent papers study this problem in 2D and show how to compute the simplicial depth in $O(n \log n)$ time [23, 26, 31]. This running time is optimal [4]. In 2003, Burr et al. [12] presented an alternate definition for the simplicial depth to overcome some unpleasant behaviors that emerge when dealing with degeneracies. Since we will be dealing with approximations, we will assume general position and thus avoid issues with degeneracy. In 3D, the first non-trivial result offered the bound of $O(n^2)$ [23] but it was flawed; fortunately, the running time of $O(n^2)$ could still be obtained with proper modifications [21]. The same authors presented an algorithm with running time of $O(n^4)$ in 4D. For dimensions beyond 4 there seems to be no significant improvements over the trivial $O(n^{d+1})$ brute-force solution. Furthermore, it is natural to conjecture that computing the simplicial depth should require $\Omega(n^{d-1})$ time: given a set P of n points, it is generally conjectured that detecting whether or not $d+1$ points lie on a hyperplane requires $\Omega(n^d)$ time [22] and this conjecture would imply that detecting whether d points of P and a fixed point q lie on a hyperplane should require $\Omega(n^{d-1})$ time. This is one motivation to consider the approximate version of the problem. In fact, Burr et al. [12] have already expressed interest in computing an approximation to the simplicial depth and they propose a potential approach, although without any worst-case analysis [11]. In 2007, Bagchi et al. [8] presented a data structure for the two-dimensional case: using $O(n \operatorname{polylog} n)$ preprocessing time, they can additively ε-approximate the simplicial depth of a given query point in $O(1)$ time.

Here, we only consider relative approximation; additive approximation (with additive error of εn^{d+1}) can be obtained using ε-nets and ε-approximations (see [16, 8] for more details).

Another motivation for computing a relative approximation comes from applications in outlier removal. Intuitively, statistical depth measures how deep a point is embedded in the data cloud with outliers corresponding to points with small values of depth. In such applications, if a small relative error of $(1+\varepsilon)$ is tolerable, then faster outlier removal can be possible using approximations.

There are several notions of data depth for which approximation and related computational problems have been considered. Aronov and Har-Peled [6] describe general techniques to attack approximation problems related to various notions of depth including finding an approximately deepest point in an arrangement of pseudodisks, approximating the depth of a query point in an arrangement of pseudodisks, approximate halfspace range counting and an approximate version of linear programming with violations which both can be formulated to depth-related problems in an arrangement of halfspaces.

Approximate halfspace range counting received most of the attention [2, 3, 25] but this also renewed interest on the general study of relative approximations [7]. Continuing this line of research Afshani and Chan presented data structures to approximate the depth of a query point in an input set of points for Tukey depth in 3D and regression depth in 2D [2].

Our Results. In Sections 3 and 7, we consider the simplicial depth problem in 2 and 3 dimensions. For $d = 2$, we present a data structure of size $\tilde{O}(n)$[1] with $\tilde{O}(n)$ preprocessing that returns the relative ε-approximation of the simplicial depth of a query point q with high probability in $\tilde{O}(1)$ time, where $\varepsilon > 0$ is an arbitrary constant. In Section 6, we consider the simplicial depth problem in arbitrary but fixed dimensions. We present two algorithms that each compute a $(1+\varepsilon)$-approximation, however with different worst-case scenarios. A combination of these strategies gives an algorithm that returns a $(1+\varepsilon)$-approximation of the simplicial depth with high probability in $\tilde{O}(n^{d/2+1})$ time. Finally, we show in Section 9 that computing the simplicial depth becomes #P-complete and W[1]-hard with respect to the parameter d if the dimension is part of the input.

Technical Difficulties. One standard technique to approximate various geometric measures is the use of uniform random samples combined with Chernoff type inequalities. Often uniform random sampling enables us to approximate the depth with high probability if the depth lies in a certain range. This property is exploited by building a hierarchy of random samples that cover all possible ranges of data depth (see [2, 6, 13, 25]). However, these existing techniques seem insufficient to approximate the simplicial depth. One

[1] The $\tilde{O}(\cdot)$ notation hides a constant number of polylogarithmic factors of n, e.g., $\log n = \tilde{O}(1)$.

particular troubling situation is depicted in Figure 1: in this configuration no high probability bound can be achieved for the simplicial depth of q in a uniform random sample despite the fact that q is far from being an outlier (it has Tukey depth $\Theta(n^{1/3})$ and simplicial depth $\Theta(n^2)$). This poses serious problems for all the previous techniques including the general techniques of Aronov and Har-Peled [6], Afshani and Chan [2] and Kaplan and Sharir [25]. In fact, it can be seen that the definition of simplicial depth prevents us from using any technique which depends on a Chernoff-type inequality.

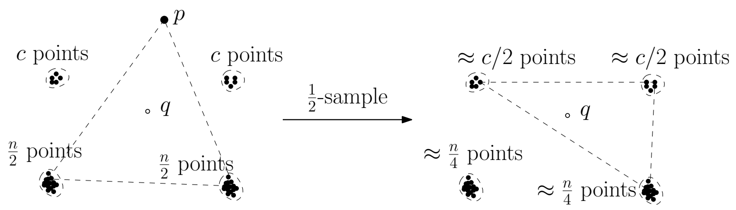

Fig. 1: (a) The point q has simplicial depth $\Theta(n^2)$ for $c = O(1)$. (b) The random sample misses point p and the $n^2/4$ triangles that shared p as a vertex; the simplicial depth of q is now $O(n^{4/3})$.

2 Structural Theorems and Preliminaries

In this section, we provide non-algorithmic results and review properties of simplicial depth which can be of independent interest. In the following sections, these results are used extensively to approximate the simplicial depth.

2.1 Notation

Throughout this paper, we denote the input set of n points in $\mathbb{R}^d$ (for a constant d) by P and the query point by q. For a given set of points P, the Tukey depth of a point q with respect to P, denoted by $\tau_P(q)$, is the minimum number of points of P that is contained in a halfspace h with $q \in h$. We will always assume the set $P \cup \{q\}$ is in general position. We denote with $\Delta_P(q)$ all subsets $P' \subseteq P$, $|P'| = d + 1$, that contain q in their convex hull. For a point $p \in \mathbb{R}^d$, let $\Delta_{p;P}(q) \subseteq \Delta_P(q)$ be the set of simplices formed by p and d points in P that contain q. We define the weight function $\omega_{q;P}(p) = |\Delta_{p;P}(q)|$ (we might omit P in the subscript if there is no danger of ambiguity). Note that we have $\sum_{p\in P} \omega_{q;P}(p) = (d + 1)\sigma_P(q)$. Given any two points v and u, we denote the line passing through them by $\overline{uv}$ and the ray from v and through u by $\overrightarrow{vu}$. We call a set S an α-sample of a set U if every element of U is chosen uniformly and randomly in S with probability

α. We will say an event X happens with high probability if $\Pr[X] \geq 1 - n^{-c}$ in which c is some (large) constant.

2.2 Bounding the Simplicial Depth with Tukey Depth

The main result of this section are tight asymptotic bounds between $\sigma_P(q)$ and $\tau_P(q)$. These results will be useful later on, but we believe they might be of independent interest as well. We start with the following easy lemma.

Lemma 2.1. *Let $P \subset \mathbb{R}^d$ be a point set and $q \in \mathbb{R}^d$. Let $S \subseteq P$, $|S| = d+1$, be a subset with $q \in \operatorname{conv}(S)$. For every point $p \in P$, there exists a unique vertex v of S such that the simplex $S \cup \{p\} \setminus \{v\}$ contains q.*

Proof. Consider the ray r starting from q in direction $q - p$. Since q is inside $\operatorname{conv}(S)$, r intersects a facet of $\operatorname{conv}(S)$. An easy calculation reveals the vertex v opposite to this facet is the vertex claimed in the lemma. Furthermore, v is unique since we assume general position. □

We call the operation used in the above lemma *swapping*. The main result of this section is the following.

Lemma 2.2. *For any point set $P \subset \mathbb{R}^d$ and any $q \in \mathbb{R}^d$, $\sigma_P(q) = \Omega(|P|\tau_P^d(q))$ and $\sigma_P(q) = O(|P|^d\tau_P(q))$. Furthermore, these bounds are tight.*

Proof. If $\tau_P(q) = 0$, then q is outside the convex hull of P and there is nothing left to prove, so assume otherwise. By Carathéodory's theorem, there exists a simplex Δ_1 formed by $d+1$ points of P which contains q. Removing Δ_1 from P reduces the Tukey depth of q by at most d and by repeating this operation we can find m disjoint subsets $\Delta_1, \dots, \Delta_m \subset P$ where $m \geq \frac{\tau_P(q)}{d}$. Let $A = \bigcup \Delta_i$. Using the Colorful Carathéodory Theorem, and following the exact same technique as Bárány [9], it follows that $\sigma_A(q) = \Omega(m^{d+1}) = \Omega(\tau_P^{d+1}(q))$. We assume $|A| \leq |P|/2$, otherwise $\Delta_A(q)$ already contains $\Omega(|P|^{d+1})$ simplices and there is nothing left to prove. Using Lemma 2.1, for any $p \in P \setminus A$ and for every simplex $\Delta \in \Delta_A(q)$, we can create another simplex Δ' by adding p and removing some other point. This way, we can create $\sigma_A(q)$ simplices for every point p. However, some of these simplices could be identical. The main observation is that any simplex Δ', can be obtained through at most $|A|$ different ways. Thus, it follows that there are at least $\Omega(|\Delta_A(q)|/|A|) = \Omega(\tau_P^d(q))$ distinct simplices $\Delta_{p,A}$ with p as a vertex and no other point from $P \setminus A$. For two points p and p' in $P \setminus A$, the sets $\Delta_{p,A}$ and $\Delta_{p',A}$ are disjoint. Hence, in total we have produced $\Omega(|P|\tau_P^d(q))$ distinct simplices.

To prove the upper bound, consider a halfspace h that passes through q and contains $\tau_P(q)$ points. Every simplex containing q must have at least one point from $h \cap P$. The maximum number of such simplices is at most $|P|^d\tau_P(q)$.

To demonstrate the tightness of the upper bound, consider a simplex Δ and a point q inside it. Replace one vertex of the simplex with a cluster of m points placed closely to each other and replace all the remaining vertices with clusters of n points. The resulting point set P_1 will contain $\Theta(n)$ points with $\sigma_{P_1}(q) = mn^d$ and $\tau_{P_1}(q) = m$. The tightness of the lower bound is realized by a very similar construction but using clusters of size m at every vertex except one, and using a cluster of size n at the remaining vertex. The resulting point set P_2 will contain $\Theta(n)$ points with $\sigma_{P_2}(q) = m^d n$. $\square$

2.3 Properties of Random Samples

A big obstacle to approximating the simplicial depth is that it is not easy to estimate the simplicial depth from a random sample: the probability that a simplex Δ survives in an α-sample of the point set is exactly α^{d+1} but these probabilities can be highly dependent for different simplices. Because of this, we need proper tools to deal with such dependence. One such tool is Azuma's inequality.

Azuma's Inequality. *Suppose $\{X_k\}$ is a martingale with the property that $|X_i - X_{i-1}| \leq c_i$. Then*

$$\Pr[|X_n - X_0| \geq t] \leq e^{-\frac{t^2}{2\sum_{k=1}^{n} c_k^2}}.$$

The following lemma is our main tool for estimating the simplicial depth from random samples.

Lemma 2.3. *Let $P \subset \mathbb{R}^d$ be a point set of size n and $q \in \mathbb{R}^d$ an arbitrary point. There exists a universal constant C such that for any parameter $\varepsilon > 0$ the following holds. Set $M = C^{-1}\varepsilon^2\sigma_P(q)/\log n$ and pick P_{large} as a subset of P that includes all points p with $\omega_{q;P}(p) \geq M$. Build a sample $S \subseteq P$ by adding a $1/2$-sample of $P \setminus P_{large}$ to P_{large}. Then, the event $|\mathbb{E}(\sigma_S(q)) - \sigma_S(q)| \geq \varepsilon\sigma_P(q)$ holds with high probability.*

Proof. Let $P' = P \setminus P_{\text{large}}$ and let S' be a $1/2$-sample of P'. By construction, we have $S = S' \cup P_{\text{large}}$. Let $p_1, \ldots, p_{n'}$ be an ordering of the points of P'. We build a martingale by revealing presence or absence of points of P' in S' in this order. Define $x_i = 1$ if p_i is sampled in S' and 0 otherwise. Let X_i be the random variable corresponding to the expected value of $\sigma_S(q)$ in which the values of $x_1, \ldots, x_i$ have been revealed. That is, the expectation is taken over $x_{i+1}, \ldots, x_{n'}$. According to this definition, $X_{n'}$ is equal to $\sigma_S(q)$, since we have revealed all the points in our sample, while X_0 is equal to $\mathbb{E}(\sigma_S(q))$, since we have revealed nothing.

The sequence $X_0, \ldots, X_n$ has the martingale property:

$$\mathbb{E}(X_{i+1}|X_1, \cdots, X_i) = \mathbb{E}(X_{i+1}|X_i)$$

and furthermore, the difference between X_{i+1} and X_i is the knowledge of x_{i+1}. However, a simple calculation reveals that the contribution of a simplex with p_{i+1} as a vertex is exactly the same in both $\mathbb{E}(X_{i+1}|X_i)$ and $\mathbb{E}(X_i)$. Thus, $\mathbb{E}(X_{i+1}|X_i) = \mathbb{E}(X_i)$.

Let $c_i = |X_i - X_{i-1}|$ as in Azuma's inequality. We now show that c_i is at most $\omega_{q;P}(p_i) < M$. If p_i is not sampled, then any simplex with p_i as a vertex will not survive and thus their contribution to the expected value of $\sigma_S(q)$ will be zero, a decrease of at most $\omega_{q;P}(p_i)$ in the expected value. So assume p_i is part of the sample, and consider a simplex Δ that contains q and is composed of p_i, t points $p'_1, \cdots, p'_t \in \{p_{i+1}, \cdots, p_{n'}\}$, and $d-t$ point $p''_1, \cdots, p''_{d-t} \in \{p_1, \cdots, p_{i-1}\}$. If any of the points $p''_1, \cdots, p''_{d-t}$ have not been sampled, then the contribution of Δ to the expected value of $\sigma_S(q)$ is zero so assume we have revealed that all these points have been sampled in S. The contribution of Δ to the expected value of $\sigma_S(q)$ before revealing that $x_i = 1$ was exactly $2^{-(t+1)}$, equal to the probability that we sample all the points $p'_1, \cdots, p'_t$ and p_i. After revealing $x_i = 1$, this contribution increases to 2^{-t}. Clearly, over all simplices with p_i as a vertex, this increase is at most $\omega_{q;P}(p_i)$. Since $p_i \notin P_{\text{large}}$, the magnitude of the change is at most M in both cases.

As discussed, $X_{n'} = \sigma_S(q)$ and $X_0 = \mathbb{E}(\sigma_S(q))$. Note that X_0 is not a random variable and that we have $\sum_{i=1}^{n'} c_i \le \sum_{i=1}^{n'-1} \omega_{q;P}(p_{i+1}) = O(\sigma_P(q))$. By Azuma's inequality we have

$$\Pr[|X_{n'} - X_0| \ge \varepsilon\sigma_P(q)] \le e^{-\frac{\varepsilon^2\sigma_P^2(q)}{\sum c_i^2}} \le e^{-\frac{\varepsilon^2\sigma_P^2(q)}{M\sum c_i}} \le e^{-\Omega\left(\frac{\varepsilon^2\sigma_P(q)}{M}\right)} = e^{-\Omega\left(\frac{\varepsilon^2\sigma_P(q)\log n}{C^{-1}\varepsilon^2\sigma_P(q)}\right)}.$$

The lemma follows by picking C large enough. $\square$

The above lemma can be used to reduce the problem of computing $\sigma_P(q)$ to computing the simplicial depth of q with the respect to a set S of roughly half the size of P. Furthermore, the value of $\mathbb{E}(\sigma_S(q))$ is directly tied to the value of $\sigma_P(q)$: any simplex $\Delta \in \Delta_P(q)$ that contains t points from the set P' and $d+1-t$ points from P_{large} contributes exactly 2^{-t} to $\mathbb{E}(\sigma_S(q))$.

3 Approximating the Simplicial Depth in 2 Dimensions

The main result of this section is a data structure of near-linear size that can answer approximate simplicial depth queries in polylogarithmic time. Later, it will be used to get an almost-optimal algorithm for approximating the simplicial depth in 3D. The problem can be stated as a triangle counting problem: given a set of n points P in the plane, build a data structure capable of approximating the number of triangles formed by points of P that contain a query point q.

Let S be a $\frac{1}{2}$-sample from P. Consider the simple and easy case when P_{large} is empty. Then, we have $\mathbb{E}(\sigma_S(q)) = \sigma_P(q)/8$ by linearity of expectation and the observation that any triangle made by points of P containing q survives with probability $1/8$. By Lemma 2.3, we can conclude that a recursively computed approximation for $\sigma_S(q)$ is with high probability very close to $\sigma_P(q)/8$. If P_{large} is not empty, then it can only contain polylogarithmically many points (as each point contributes a significant amount to the simplicial depth) and thus we need to keep track of a "few" points. The biggest challenge, however, is finding the subset P_{large}. This is done with the following lemma. Unfortunately its proof does not seem to be easy and in fact it requires overcoming many technical steps and combining shallow cuttings with various observations regarding the geometry of planar points. We ultimately reduce the problem to instances of orthogonal range reporting problem in eight(!) dimensional space, which fortunately can be solved with $\tilde{O}(n)$ space and $\tilde{O}(1)$ query time. The proof is given in Section 4.

Lemma 3.1. *Let P be set of n points and let $S'_0 = P, S'_1, \dots, S'_r$ in which S'_{i+1} is a $\frac{1}{2}$-sample from S'_i. There exists a data structure of size $\tilde{O}(n)$, such that the following holds with high probability. Given j and a query point q, define $M = C^{-1}\varepsilon^2\sigma_{S'_j}(q)/\log n$, where C is a constant and $\varepsilon > 0$ is a fixed parameter. If $\tau_{S_j}(q) = \Omega(\varepsilon^{-2}\log^3 n)$, then the data structure can find the set $P_{large} \subset S_j$ that contains all the points p with $\omega_{q;S_j}(p) \geq M$ in $\tilde{O}(1)$ time. The data structure can be built in $\tilde{O}(n)$ expected time.*

Also, the data structure defines $O(n)$ canonical halfplanes, such that P_{large} lies inside a halfplane $h(q)$ that contains q and $\tilde{O}(\tau_{S_j}(q))$ points of S_j. $h(q)$ only depends on q and not j.

We also need the following lemma. For the proof, see Section 5.

Lemma 3.2. *Given a parameter $\varepsilon > 0$, we can store a set S of n points in a data structure of size $\tilde{O}(n)$ that can answer the following queries. Given a query point q, a halfplane h with q on its boundary containing $\tilde{O}(\tau_S(q))$ points of S, and a subset of $R \subset S \cap h$, we approximate the total number of triangles that contain q and include at least one point from R in $\tilde{O}(\varepsilon^{-2}|R|^2)$ time and with additive error of at most $\varepsilon\sigma_S(q)$.*

In the rest of this section, we outline our solution to approximate the simplicial depth of a query point. Consider a series of random samples $S'_0 = P, S'_1, \dots, S'_r$ in which S'_{i+1} is a $1/2$-sample from S'_i and $|S_r| = \tilde{O}(1)$. We store each sample S'_i in the data structure from Lemma 3.2. Furthermore, we store the sampling sequence sampling sequence $S'_0, \cdots, S'_r$ in the data structure from Lemma 3.1.

We use a recursive approach where Lemmata 2.3, 3.1, and 3.2 are our bread and butter: during step i of the query algorithm, we are given a set

$S^{(i)}_{\text{large}} \subset R(q) \cap P$ containing $\tilde{O}(1)$ points, such that $S^{(i)}_{\text{large}} \cap S'_i = \emptyset$. The goal is to compute $\sigma_{S_i}(q)$ where $S_i = S'_i \cup S_{\text{large}}$. Initially, $S^{(0)}_{\text{large}} = \emptyset$ and $S'_0 = P$.

Our strategy will be to recursively compute the simplicial depth. We will assume our recursion returns a relative $(1+\delta)$-approximation of $\sigma_{S_{i+1}}(q)$. By using Lemmata 2.3, 3.1, and 3.2 with parameter ε, we can return a value that is a relative $(1+\delta+O(\varepsilon))$-approximation of $\sigma_{S_i}(q)$. We set $\varepsilon = 1/\delta$ such that at the top of the recursion we end up with a relative $(1+O(\delta))$-approximation. Now we present the details.

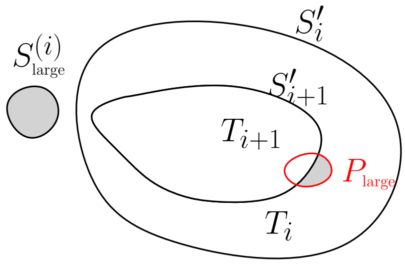

Fig. 2: S'_{i+1} is a 1/2-sample of S'_i. $P_{\text{large}} \subset S'_i$ is obtained through Lemma 3.1 and it is shown in red. $S_i = S^{(i)}_{\text{large}} \cup S'_i$, $T_{i+1} = S'_{i+1} \setminus P_{\text{large}}$, $T_i = S'_i \setminus P_{\text{large}}$, and the greyed areas represent $S^{(i+1)}_{\text{large}}$.

Approximating $\sigma_{S_i}(q)$. Let $\varepsilon = \delta/\log n$. First, observe that if $\tau_{S_i}(q) = \tilde{O}(1)$, then we can directly approximate $\sigma_{S_j}(q)$ using Lemma 3.2 by setting $R = R(q) \cap S_j$ in Lemma 3.2. In the rest of this proof, we assume that $\tau_{S_i}(q) = \Omega(\varepsilon^{-2}\log^3 n)$.

See Figure 2 for a Venn diagram of the various subsets involved here. We set $M = C^{-1}\varepsilon^2\sigma_{S'_i}(q)/\log n$ and using Lemma 3.1, we find the set $P_{\text{large}} \subset S'_i$. Let $T_{i+1} = S'_{i+1} \setminus P_{\text{large}}$, $T_i = S'_i \setminus P_{\text{large}}$, and $S^{(i+1)}_{\text{large}} = (P_{\text{large}} \setminus S'_{i+1}) \cup S^{(i)}_{\text{large}}$. Clearly, $S^{(i+1)}_{\text{large}} \cap S_{i+1} = \emptyset$, and $S^{(i+1)}_{\text{large}} \subset R(q) \cap P$ so we can recurse. Assume we obtain a value Y that is a relative $(1+\delta)$ approximation of $\sigma_{S_{i+1}}(q)$. Note that T_{i+1} is a 1/2-sample of T_i. From the definition of P_{large} and by Lemma 2.3, we can conclude that with high probability

$$|\mathbb{E}[\sigma_{T_{i+1}\cup P_{\text{large}}}(q)] - \sigma_{T_{i+1}\cup P_{\text{large}}}(q)| \le \varepsilon\sigma_{S'_i}(q). \tag{1}$$

By Lemma 3.2, we can approximate the number of triangles, m, that contain at least one point from $S^{(i)}_{\text{large}}$ with additive error $\varepsilon\tau_{S_{i+1}}(q)$. This combined with Y gives an approximate value of $\sigma_{T_{i+1}\cup P_{\text{large}}}(q)$ with additive error $(O(\varepsilon)+\delta)\tau_{S_{i+1}}(q)$. Combined with (1), this gives an estimate for $\mathbb{E}[\sigma_{T_{i+1}\cup P_{\text{large}}}(q)]$ with additive error $(O(\varepsilon)+\delta)\tau_{S_i}(q)$. Let n_i, $0 \le i \le 3$, be the number of triangles containing q that have i points from the set P_{large} and $3-i$ points from T_i. Clearly, $\sigma_{S'_i}(q) = n_0 + n_1 + n_2 + n_3$ and $\mathbb{E}[\sigma_{T_{i+1}\cup P_{\text{large}}}(q)] =$

$n_0/8+n_1/4+n_2/2+n_3$. Using Lemma 3.2, we approximate $n_1+n_2+n_3$ with additive error $\varepsilon\tau_{S_i}(q)$. However, n_2 and n_3 are negligible compared to our error margins: by Lemma 2.2, $\sigma_{S_i}(q) = \Omega(|S_i|\tau^2_{S_i}(q))$ but n_2 can be at most $|P_{\text{large}}|^2|S_i|$ and n_3 can be at most $|P_{\text{large}}|^3$. Since $\tau_{S_i}(q) = \Omega(\varepsilon^{-2}\log^3 n)$, it follows that we can ignore n_2 and n_3 in our calculations and that Lemma 3.2 reveals an approximation of n_1. With these observations, we can obtain an approximation $\sigma_{S'_i}(q)$ with additive error $(O(\varepsilon)+\delta)\tau_{S_i}(q)$. Combining this with an approximation of m yields an approximation of $\sigma_{S_i}(q)$ with additive error $(\delta+O(\varepsilon))\sigma_{S_i}(q)$. Observe that the error factor in our additive term has worsened from δ (regarding Y and $\sigma_{S_{i+1}}(q)$) to $\delta+O(\varepsilon)$ (regarding $\sigma_{S_i}(q)$). However, we have $\varepsilon = \delta/\log n$ and there are at most $O(\log n)$ recursion steps. Thus at the top of the recursion (that is for $\sigma_P(q)$), we obtain a $(1+O(\delta))$ approximation factor, as claimed.

Theorem 3.3. *It is possible to preprocess a point set $P \subset \mathbb{R}^2$ of n points in $\tilde{O}(n)$ expected time using $\tilde{O}(n)$ space such that, given a query point q, a relative ε-approximation for the simplicial depth of q can be found in $\tilde{O}(1)$ expected time for any arbitrary fixed constant $\varepsilon > 0$ with high probability.*

4 Proof of Lemma 3.1

Lemma 4.1. *Let $P \subset \mathbb{R}^2$ be a set of n points, $q \in \mathbb{R}^2$ an arbitrary point and h a halfplane containing at most $C'\tau_P(q)$ points with q at its boundary. Consider the line $\overline{pq}$ for a point $p \in h$ and assume it partitions P into two sets of sizes n_1 and n_2. If $n_1, n_2 \geq 2C'\varepsilon^{-1}\tau_P(q)$ then n_1n_2 is a relative $(1+\varepsilon)$-approximation of $\omega_{q;P}(p)$.*

Proof. The number of triangles containing q and involving p is at least $(n_1-m)(n_2-m)$ and at most n_1n_2 where $m \leq C'\tau_P(q)$ is the number of points in h. The lemma follows by a simple calculation and observing that n_1n_2 is minimized when n_1 or n_2 is $2C'\varepsilon^{-1}\tau_P(q)$. □

We also use shallow cuttings which we state in two dimensions. For a set of lines H in the plane, the *level* of a point p is the number of points that pass below p. Given an integer k, the k-level of H is the closure of all the points that have level exactly k. The $(\leq k)$-level is defined as the closure of all the points with level at most k.

Theorem 4.2. *Let H be a set of n lines in the plane and k be a given parameter $1 \leq k < n/2$. We can find a convex polygonal chain C of size $O(n/k)$ such that it lies above the k-level of H, the level of every vertex of C is $O(k)$. The cutting can be constructed in $O(n\log n)$ time.*

Proof. Matoušek [30] proved that one can cover the $(\leq k)$-level with $O(n/k)$ triangles such that there are at most $O(k)$ lines passing below each triangle. Chan [13] observed that we can consider the convex hull of the triangles; the number of lines passing below the convex hull is only increasing by $O(k)$.

Ramos [29] offered a randomized $O(n \log n)$ construction in 1999 and recently Chan and Tsakalidis have shown the same running time can be achieved with a deterministic algorithm [15]. □

We will also be working with point-line duality in the plane. This transformation, maps a line ℓ passing below (reps. above) a point p to point $\overline{\ell}$ that lies below (resp. above) the line $\overline{p}$.

Lemma 4.3. *Let P be set of n points and let $S'_0 = P, S'_1, \ldots, S'_r$ in which S'_{i+1} is a $\frac{1}{2}$-sample from S'_i. There exists a data structure of size $\tilde{O}(n)$, such that the following holds with high probability. Given j and a query point q, define $M = C^{-1}\varepsilon^2\sigma_{S'_j}(q)/\log n$, where C is a constant and $\varepsilon > 0$ is a fixed parameter. If $\tau_{S_j}(q) = \Omega(\varepsilon^{-2}\log^3 n)$, then the data structure can find the set $P_{large} \subset S_j$ that contains all the points p with $\omega_{q;S_j}(p) \geq M$ in $\tilde{O}(1)$ time. The data structure can be built in $\tilde{O}(n)$ expected time.*

Also, the data structure defines $O(n)$ canonical halfplanes, such that P_{large} lies inside a halfplane $h(q)$ that contains q and $\tilde{O}(\tau_{S_j}(q))$ points of S_j. $h(q)$ only depends on q and not j.

Proof. We will describe the data structures incrementally. Consider the set $\overline{P}$ of n lines dual to P. Let $k_i = 2^i$, $1 \leq i < \log n$. First, by Matoušek's shallow cutting theorem (Theorem 4.2), for each k_i, we build a shallow cutting L_i for the $(\leq k_i)$-level of $\overline{P}$, that is a convex polygonal chain of size $O(n/k_i)$ that lies between k_i-level and $O(k_i)$-level of $\overline{P}$. Each vertex v of the polygonal chain L_i defines a canonical halfplane in primal space (the region below $\overline{v}$). For each S_j, the subset of S_j that lies inside a canonical halfplane is called a *canonical set.* We also do the same for the $(\geq k_i)$-level (to obtain cuttings U_i). By Theorem 4.2, each chain L_i has $O(n/k_i)$ vertices and thus creates $O(n/k_i)$ canonical sets. Furthermore, each canonical set contains $O(k_i)$ points. Thus, the total size of canonical sets on P is $O(n \log n)$ which is also an upper bound for the total size of the canonical sets on each S_i. This means, the total size of canonical sets, over all indices $j, 1 \leq j \leq \log n$ is $O(n \log^2 n)$. Furthermore, a standard application of the Chernoff bound yields that L_i is below $(\leq \Theta(k_i \log n/2^j))$-level and above $(\leq (\Theta(k_i/(2^j \log n)))$-level of S_j, with high probability (similarly for U_i); let's call this the *level property.* In the rest of this proof, we will build a separate data structure for each S_j. There are $O(\log n)$ different indices j and thus this would only blow up the space by a $\log n$ factor. Let $S = S_j$ be the subset we are currently working with. Given a query point q, the goal is to report a subset $P_{\text{large}} \subset S$ that contains all the points p with $\omega_{q;S}(p) \geq M$.

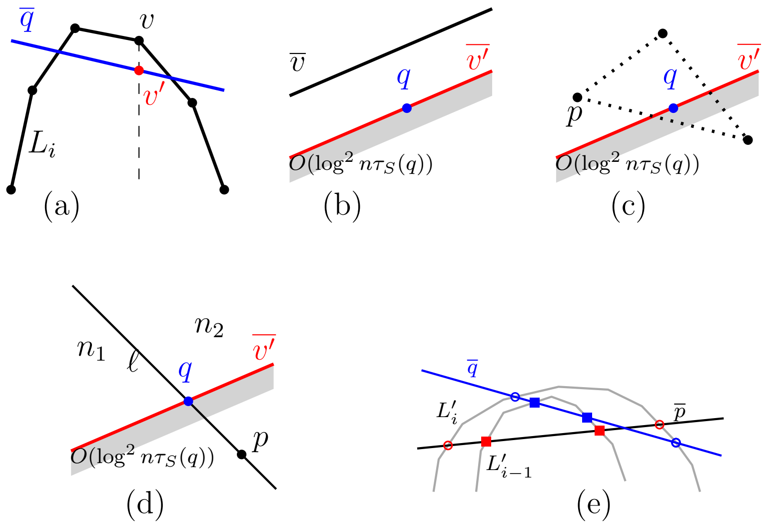

Fig. 3: (a) In dual space, $\overline{q}$ passes below a vertex v of a shallow cutting. The set of lines below v are considered a *canonical set*. (b) In primal space, the region below $\overline{v}$ is a canonical halfplane. We can draw a line $\overline{v'}$ parallel to v from q. (c) Any point that is above $\overline{v'}$ cannot create M triangles that contain q since it is forced to pick a point from below $\overline{v'}$ and there are only few such points. (d) p lies below $\overline{v}$. The line ℓ is defined by connecting p and q. There are n_1 points below the line ℓ and n_2 points above it. (e) If a line intersects the convex chain L'_i, then it creates an interval on its boundary. We can tell if two lines intersects between chains L'_{i-1} and L'_i by examining the corresponding intervals that they define.

The given query point q, corresponds to a query line $\overline{q}$ in the dual space. If i is the smallest index such that $\overline{q}$ intersects L_i or U_i, then $k_i/2^j$ is an approximation of $\tau_S(q)$ up to $O(\log^2 n)$ factor, by the level property. W.l.o.g, assume $\overline{q}$ intersects L_i and thus passes below a vertex $v \in L_i$ (the other case when $\overline{q}$ intersects U_i can be found in an analogous way and by building analogous data structures). Let v' be the point on $\overline{q}$ the lies directly below v (Figure 3(a)). In the primal space, v' corresponds to a line that goes through q and has $O(\log^2 n\tau_S(q))$ points below it, with high probability (by definition of $\tau_S(q)$, there are at least $\tau_S(q)$ points below $\overline{v'}$ as well). Furthermore, v' is parallel to a canonical halfplane that is the dual of v (see Figure 3(a,b)). v' defines $h(q)$.

We now claim, P_{large} has to be a subset of points of S that are below $\overline{v'}$. See Figure 3(c). Consider a point p that is not below v'; any triangle that contains q and includes p, must include a point below v'. Remember that v' has only $O(\log^2 n\tau_S(q))$ points below it. This means, $\omega_{q;S}(p) = O(|S|\log^2 n\tau_S(q))$. By Lemma 2.2, we have $\sigma_S(q) = \Omega(|S|\tau_S^2(q))$. Note that we need to answer queries for when $\tau_S(q) = \Omega(\varepsilon^{-2}\log^3 n)$, namely, when $\tau_S(q) > C\varepsilon^{-2}\log^3 n$, so we have $\sigma_S(q) = \Omega(|S|\varepsilon^{-2}\log^3 n\tau_S(q))$. This means $\omega_{q,S}(p) < M$.

Now that we have established P_{large} has to be a subset of S below $\overline{v'}$, we turn our attention to building the proper data structures to find it.

Consider a canonical set that contains the subset of S below the line $\overline{v}$; denote this canonical set by $v^{\downarrow}$ and let $|v^{\downarrow}| = m$. We can build $O(m)$ canonical *subgroups*, where each subgroup is a subset of $v^{\downarrow}$ such that the points below any line $\overline{v'}$ parallel to $\overline{v}$ can be expressed as the union of at most $\log m$ canonical subgroups: this is done by projecting the points of $v^{\downarrow}$ onto a line perpendicular to $\overline{v}$, and then building a balanced binary tree on the resulting one-dimensional point set; each node of the balanced binary tree defines a canonical subgroup. The total size of the canonical subgroups created on $v^{\downarrow}$ is $O(m\log m)$.

Remember that our goal was to find all the points p below $\overline{v'}$ with $\omega_{q;S}(p) \geq M$. We can now use the canonical subgroups, since there $O(\log m)$ subgroups that cover all the points below $\overline{v'}$; we can query each such subgroup independently to find the subset of P_{large} that lies in that subgroup; this would only blow up the query time by a $\log n$ factor. Furthermore, remember that the total size of all canonical sets was $O(n\log^2 n)$ and thus the total size of all canonical subgroups is $O(n\log^3 n)$. Thus, we can afford to build a separate data structure for each canonical subgroup.

Consider a canonical subgroup $G \subset v^{\downarrow}$ that contains g points. Our new goal is to find all points $p \in G$ such that $\omega_{q;S}(p) \geq M$. We claim to approximate $\omega_{q;S}(p)$ it suffices to draw the line $\ell = pq$ and then multiply the number of points of S that lie at either side of ℓ. See Figure 3(d). Let $n_1 = |\ell^- \cap S|$ and $n_2 = |\ell^+ \cap S|$ where ℓ^+ and ℓ^- corresponds to the halfplane above and below ℓ. W.l.o.g, assume $n_1 \geq n_2$. Consider a point p with $\omega_{q;S}(p) \geq M$. We have

$$M = \frac{C^{-1}\varepsilon^2\sigma_S(q)}{\log n} = \Omega\left(\frac{C^{-1}\varepsilon^2|S|\tau_S^2(q)}{\log n}\right)$$

and thus we must have

$$\Omega\left(\frac{C^{-1}\varepsilon^2|S|\tau_S^2(q)}{\log n}\right) \leq \omega_{q;S}(p) \leq n_1 n_2 \leq |S|n_2$$

and thus

$$n_2 = \Omega\left(\frac{\varepsilon^2\tau_S^2(q)}{\log n}\right).$$

Let r be the number of points below $\overline{v'}$. On the other hand, we have

$$\omega_{q;S}(p) \geq (n_1 - r)(n_2 - r) \geq n_1 n_2 - |S|r.$$

Since $r = O(\log^2 n\tau_S(q))$, it follows that $n_2/r = \Omega\left(\frac{\varepsilon^2\tau_S(q)}{\log^3 n}\right) = \Omega(\varepsilon^{-1})$ and thus $n_1 n_2$ is a constant factor approximation of $\omega_{q;S}(p)$. This in turn implies

$|S|n_2$ is also a constant factor approximation of $\omega_{q;S}(p)$. This is the motivation for defining the following concept: For a point p below $\overline{v'}$, we call the minimum number of points of S_i at either side of line $\overline{pq}$ its *dissection value with respect to q and S_i* (or its dissection value for short). In Figure 3(d), n_2 is the dissection value of p, if $n_1 \geq n_2$.

Thus, our goal can be further reduced to the following, that we call *dissection reporting*: For a canonical subgroup $G \subset v^{\downarrow}$ that contains g points, build a data structure that can answer the following: given a query point q, a threshold t, and a line $\overline{v'}$ parallel to $\overline{v}$ such that G lies below $\overline{v'}$, find all points in G with dissection value larger than t. To find the part of P_{large} in G, we simply set $t = \Omega(M/|S|)$ for an appropriate constant hiding in the $\Omega(\cdot)$ notation; as we observed, the dissection value times $|S|$ is a constant approximation of $\omega_{q;S}(p)$ so this way, we can find a subset that contains all the point in $P_{\text{large}} \cap G$. However, we will report a few extra points as well; nonetheless, we can observe that the number of extra points reported is only a constant factor larger since every point that gets reported contributes a lot to the simplicial depth of q and there cannot be too many such points.

To solve the dissection reporting problem, let $\overline{S}$ be the set of lines dual to S. We build a shallow cutting L'_i for the $(\leq k_i)$-level of $\overline{S}$ (resp. U'_i for the $(\geq k_i)$-level of $\overline{S}$), for $i = c^i, 1 \leq i \leq t = O(\log|S|)$ for a large enough constant c. L'_i is a convex polygonal chain and if c is set large enough, then L'_i lies between the k_i-level and the ck_i-level of S. This means that the polygonal chains L'_i will be non-intersecting and L'_{i-1} is contained inside L'_i. Consider a line $\overline{p}$ in dual space (see Figure 3(e)). If $\overline{p}$ intersects L'_i, then it creates an interval on L'_i, marked by the two intersection points of $\overline{p}$ with L'_i. Let's call this $I_i(\overline{p})$; note that the end points of this interval come from a one dimensional domain which can be parameterized in various different ways, such as using polar angles from a point inside each polygonal chain. Consider $I_i(\overline{p})$ for every point $p \in G$ as well as the interval $I_i(\overline{q})$ defined by the line dual to the query point. We say two intervals I_1 and I_2 intersect if they have a point in common and none of the intervals fully contains the other. Consider an index i such that $I_i(\overline{q})$ intersects the interval $I_i(\overline{p})$ but $I_{i-1}(\overline{q})$ does not intersect the interval $I_{i-1}(\overline{p})$. This means that the intersection point of the lines $\overline{q}$ and $\overline{p}$ is between L'_{i-1} and L'_i (see Figure 3(e)). Imagine for the upper chains, the intersection point of the lines $\overline{q}$ and $\overline{p}$ is between U'_{j-1} and U'_j where $j > i$. This implies that the dissection value of line pq is $\Theta(k_i)$.

Motivated by the observation in the above paragraph, we do the following. For each pair of indices (i, j), we create a data structure that is capable of finding all the lines $\overline{p} \in G$ that the intersection point of $\overline{p}$ and $\overline{q}$ lies between L'_{i-1} and L'_i and also between U'_{j-1} and U'_j. If we do this, then we can perform dissection reporting by issuing $O(\log^2 n)$ such queries: we first query (t, t), then query $(t, t-1)$ and $(t-1, t)$ and so on. Specifically, to find points with dissection value $\Theta(k_i)$ we query $(t, i), (t-1, i), \cdots, (i, i)$ as well as $(i, t), (i, t-1), \cdots, (i, i)$.

It thus remains to show how a query pair (i, j) is answered. For every point $p \in G$, we create a tuple of four input intervals corresponding to the intersection of $\overline{p}$ with L'_{i-1}, L'_i, U'_{j-1}, and U'_j. The query also defines a tuple of four intervals in a similar fashion. The goal is to store the input tuples of intervals in a data structure such that given a query tuple of intervals we can find all the input interval tuples such that they guarantee $\overline{p}$ and $\overline{q}$ intersect between L'_{i-1} and L'_i and also between U'_{j-1} and U'_j. Let $([a_1, b_1], [a_2, b_2], [a_3, b_3], [a_4, b_4])$ an input tuple of four intervals. We create an eight-dimensional point $(a_1, b_1, \cdots, a_4, b_4)$.

We now claim the problem can be solved using dominance reporting in eight dimensional space. In dominance reporting, a point $p \in \mathbb{R}^d$ is said to *dominate* a point $q \in \mathbb{R}^d$ if any coordinates of p is greater than that of q. In dominance reporting, we are to store a set of points in a data structure such that given a point q, we can report all the points dominated by q. Observe that in our subproblem, we can also map the tuple of intervals corresponding to the query to the eight dimensional space. Here, we can express the constraints that $\overline{p}$ and $\overline{q}$ intersect between L'_{i-1} and L'_i and also between U'_{j-1} and U'_j as inequalities between respective coordinates of the eight dimensional points, meaning, the problem can be solved by building a constant number of eight dimensional dominance reporting data structures and issuing a constant number of dominance reporting queries.

Dominance reporting queries can be answered in polylogarithmic time, using data structure that needs near-linear space and preprocessing time [1, 14, 10, 17, 18] polylogarithmic time, we can find the subset of P_{large} in G using polylogarithmic space overhead and polylogarithmic query time.

Thus, in overall, our data structure will use $\tilde{O}(n)$ space and preprocessing time and can answer queries in $\tilde{O}(1)$ time. □

5 Proof of Lemma 3.2

Lemma 5.1. *Given a parameter $\varepsilon > 0$, we can store a set S of n points in a data structure of size $\tilde{O}(n)$ that can answer the following queries. Given a query point q, a halfplane h with q on its boundary containing $\tilde{O}(\tau_S(q))$ points of S, and a subset of $R \subset S \cap h$, we approximate the total number of triangles that contain q and include at least one point from R in $\tilde{O}(\varepsilon^{-2}|R|^2)$ time and with additive error of at most $\varepsilon \sigma_S(q)$.*

Proof. We recall the definition of the dissection value from the proof of Lemma 3.1: For a point $p \in R$, we call the minimum number of points of S at either side of line $\overline{pq}$ its *dissection value with respect to q and S* (or its dissection value for short). Let m be the number of points in h.

We build data structure for approximate range counting [2], and halfplane range reporting [19] on S. Using these, for every point $p \in R$ we can obtain a constant factor approximation of its dissection value.

Consider a point $p \in R$. Draw the line $\ell = pq$ and let $n_1 = |\ell^- \cap S|$ and $n_2 = |\ell^+ \cap S|$ where ℓ^+ and ℓ^- corresponds to the halfplane above and below ℓ (very similar to the situation in Lemma 3.1; see also Figure 3(d)). Assume $n_1 \le n_2$. Let Δ_p be the set of triangles that contain q and have p as one of their vertices.

As first case, assume $n_1 \ge 2\varepsilon^{-1}(m + |R|)$. At most $|S||R|$ triangles from Δ_p can have two points from R. However, Δ_p contains at least

$$(n_1 - m)(n_2 - m) \ge n_1 n_2 - |S|m$$

triangles. Observe that if $n_1 \ge 2\varepsilon^{-1}(m + |R|)$ then

$$n_1 n_2 - |S|m \ge \varepsilon^{-1}|S||R|.$$

Thus, the number of triangles in Δ_p with two points from R is negligible. So, we focus on the triangles that have exactly one point from R. This is at most $n_1 n_2$ and at least $(n_1 - m)(n_2 - m)$. Again, an easy calculation yield that $n_1 n_2 \ge \varepsilon^{-1} m|S|$ and thus $n_1 n_2$ is a relative $(1 + O(\varepsilon))$-approximation of the number of triangles in Δ_p.

Thus, it suffices to handle points with dissection value less than $2\varepsilon^{-1}(m + |R|)$. In the rest of this proof, we assume all the points in R have dissection value less than this.

As a second case, assume $\tau_S(q) = \tilde{O}(\varepsilon^{-1}|R|)$ which also implies $m = \tilde{O}(\varepsilon^{-1}|R|)$, and that the dissection value of the points in R is at most $2\varepsilon^{-1}(m + |R|) = \tilde{O}(\varepsilon^{-1}|R|)$. Let $p_1, \cdots, p_r \in R$ be all the points in R. We create the lines $\ell_1, \cdots, \ell_r$ by connecting p_i to q. Using halfplane range reporting, we can exactly compute the number of points at one side of each line ℓ_i in $\tilde{O}(\varepsilon^{-1}|R|)$ time. Over all the points in R this will take $\tilde{O}(\varepsilon^{-2}|R|^2)$ time. Given these values, we can compute the number of points that lie inside each "wedge" created by the lines $\ell_1, \cdots, \ell_r$ (see Figure 4(b)) in $\tilde{O}(\varepsilon^{-2}|R|^2)$

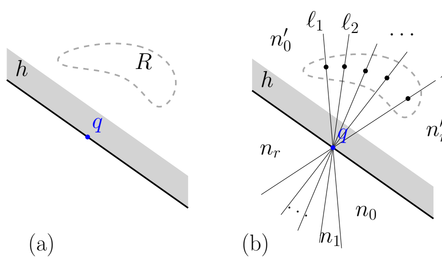

Fig. 4: (a) The input configuration for Lemma 3.2. (b) For each point $p \in R$, a line through p and q is created. This subdivides the plane into "wedges" that are ifinite triangles with q as their only vertex. By knowing the exact number of points inside each wedge, we can compute the simplicial depth in $O(r)$ time.

time. Given these values, the total number of triangles that contain q and have one or two points from the set $\{p_1, \cdots, p_r\}$ can be counted in $O(r)$ time (see [23, 26]).

Finally, we assume $\tau_S(q) \geq \varepsilon^{-1}|R| \log^2 n$. This case can be handled using similar ideas as the previous two. Here, we claim among triangles in Δ_p, the number of those that include two points from R is negligible: the number of such triangles is at most $|S||R|^2$ where at the simplicial depth of q is at least $\Omega(|S|\tau_S^2(q))$ by Lemma 2.2. We have

$$|S|\tau_S^2(q) \geq |S|\varepsilon^{-2}|R|^2 = \omega(\varepsilon^{-1}|S||R|^2)$$

so we can safely ignore triangles that contain two points from R. This limits us to triangles that have exactly one point from R. Let $p_1, \cdots, p_r \in R$ be the points in R. Again, we create the lines $\ell_1, \cdots, \ell_r$ by connecting p_i to q. Using halfplane range reporting, we can exactly compute the number of points at each side of the line ℓ_i in $\tilde{O}(\varepsilon^{-1}|R|)$ time. Over all the points in R this will take $\tilde{O}(\varepsilon^{-2}|R|^2)$ time. As before, these values enable us to compute the number of triangles containing one point from R.

Combining all these cases, we obtain a relative $(1 + O(\varepsilon)$-approximation. By scaling ε by a constant, we can obtain a relative $(1 + \varepsilon)$-approximation. The total query time is $\tilde{O}(\varepsilon^{-2}|R|^2)$. $\square$

6 Approximation in High Dimensions

In this section, we present two approximation algorithms for simplicial depth in high dimensions, each with a different worst case scenario. By combining these strategies, we obtain a constant factor approximation algorithm with $\tilde{O}(n^{d/2+1})$ running time.

6.1 Small Simplicial Depth: Enumeration

Let $P \subset \mathbb{R}^d$ be a set and $q \in \mathbb{R}^d$ a query point. If $\sigma_P(q)$ is small, a simple counting approach that iterates through all simplices $\Delta \in \Delta_P$ leads to an efficient algorithm. The key is to construct a graph that contains exactly one node per simplex $\Delta \in \Delta_P$. Then, counting can be carried out by a breadth-first search and we avoid looking at subsets of P that do not contain q in their convex hull. For this, we use the Gale transform to dualize the problem. We shortly restate important properties of the Gale transform. For more details see [28]. Let in the following $\mathbf{0}$ denote the origin.

Lemma 6.1. *Let $P = \{p_1, \ldots, p_n\} \subset \mathbb{R}^d$ be a point set with $\sigma_P(\mathbf{0}) > 0$. Then, there is a set $\bar{P} = \{\bar{p}_1, \ldots, \bar{p}_n\} \subset \mathbb{R}^{n-d-1}$ such that a $(d+1)$-subset $P' \subseteq P$ contains $\mathbf{0}$ in its convex hull iff $\bar{P} \setminus \{\bar{p}_i \mid p_i \in P'\}$ defines a facet of* $\operatorname{conv}(\bar{P})$. $\square$

Consider now the graph $G_P(q) = (V, E)$ with $V = \Delta_P$. Two simplices Δ, Δ' are adjacent iff Δ' can be obtained from Δ by swapping one point in Δ with a different point in P. We call $G_P(q)$ the *simplicial graph* of P with respect to q.

Lemma 6.2. *Let $P \subset \mathbb{R}^d$ be a set of size n. Then, $G_P(q)$ is $(n-d-1)$-connected and $(n-d-1)$-regular.*

Proof. We assume w.l.o.g. that $q = \mathbf{0}$. Let Δ, Δ' be two adjacent nodes in $G_P(q)$. Furthermore let $\bar{P}$ denote the Gale transform of P. Set $\bar{\Delta} = \{\bar{p} \mid p \in P \setminus \Delta\}$ and $\bar{\Delta}' = \{\bar{p} \mid p \in P \setminus \Delta'\}$. By Lemma 6.1, the two sets $\bar{\Delta}$ and $\bar{\Delta}'$ define facets of $\text{conv}(\bar{P})$. Since Δ and Δ' are adjacent, we have $|\Delta \cap \Delta'| = d$ and hence $|\bar{\Delta} \cap \bar{\Delta}'| = n - d - 2$. Thus, the facets defined by $\bar{\Delta}$ and $\bar{\Delta}'$ share a ridge. Hence, $G_P(q)$ is isomorph to the 1-skeleton of the polytope dual to $\text{conv}(\bar{P})$. In particular, this implies that $G_P(q)$ is $(n-d-1)$-connected. It remains to show that the graph is $(n-d-1)$-regular. Let $\Delta \in V$ be a node. Lemma 2.1 states that each of the $n-d-1$ points in $P \setminus \Delta$ can be swapped in, each time resulting in a distinct simplex. $\square$

Since $G_P(q)$ is connected, we can count the number of vertices using BFS.

Lemma 6.3. *Let $P \subset \mathbb{R}^d$ be a set of size n and $q \in \mathbb{R}^d$ a query point. Then, $\sigma_P(q)$ can be computed in $O(n\sigma_P(q))$ time.*

6.2 Large Simplicial Depth: Sampling

If the simplicial depth is large, the enumeration approach becomes infeasible. In this case we apply a simple random sampling algorithm.

Lemma 6.4. *Let $P \subset \mathbb{R}^d$ be a set and $q \in \mathbb{R}^d$ a query point. Furthermore, let $\varepsilon, \delta > 0$ be constants and let $m \in \mathbb{N}$ be a parameter. If $\sigma_P(q) \geq m$, then $\sigma_P(q)$ can be $(1+\varepsilon)$-approximated in $\tilde{O}(n^{d+1}/m)$ time with error probability $O(n^{-\delta})$.*

Proof. Let $\Delta_1, \dots, \Delta_k$ be k random $(d+1)$-subsets of P for $k = \left\lceil \frac{4\delta n^{d+1} \log n}{\varepsilon^2 m} \right\rceil$. For each random subset Δ_i, let X_i be 1 iff $q \in \text{conv}(\Delta_i)$ and 0 otherwise. We have $\mu = \mathbb{E}[\sum_{i=1}^k X_i] = k\frac{\sigma_P(q)}{n^{d+1}} = \frac{4\delta\sigma_P(q)\log n}{\varepsilon^2 m} \geq \frac{4\delta}{\varepsilon^2}\log n$. Applying the Chernoff bound, we get $\Pr[|\sum_{i=1}^k X_i - \mu| \geq \varepsilon\mu] = O(n^{-\delta})$. Thus, $\frac{n^{d+1}}{k}X$ is a $(1+\varepsilon)$-approximation of $\sigma_P(q)$ with error probability $O(n^{-\delta})$.

For $d = O(1)$, we can test in $O(1)$ whether a given $(d+1)$-subset of P contains a point in its convex hull. Hence, the running time is dominated by the number of samples. $\square$

6.3 Combining the Strategies

Theorem 6.5. *Let $P \subset \mathbb{R}^d$ be a set and $q \in \mathbb{R}^d$ a query point. Furthermore, let $\varepsilon > 0$ and $\delta > 0$ be constants. Then, $\sigma_P(q)$ can be $(1+\varepsilon)$-approximated in $\tilde{O}(n^{d/2+1})$ time with error probability $O(n^{-\delta})$.*

Proof of Theorem 6.5. We apply the algorithm from Lemma 6.3 and stop it once $n^{d/2}$ nodes of $G_P(q)$ are explored. This requires $O(n^{d/2+1})$ time. If the graph is not yet fully explored, we know $\sigma_P(q) \geq n^{d/2}$. We can now apply the algorithm from Lemma 6.4 and compute a $(1+\varepsilon)$-approximation in $\tilde{O}(n^{d/2+1})$ time with error probability $O(n^{-\delta})$. □

7 Improved Approximation in Three Dimensions

In this section, we show that the simplicial depth in 3D can be approximated in $\tilde{O}(n)$ time, which is clearly optimal up to polylogarithmic factors. The main ingredients required for our proof are the two-dimensional data structure from Section 3, Lemma 2.2, and Observation 8.1.

Theorem 7.1. *Let P be a set of n points in 3D. The simplicial depth of a given point q can be approximate in $\tilde{O}(n)$ expected time and with high probability.*

Proof. First, we find a halfspace h that passes through q and contains a subset $A \subset P$ of $\Theta(\tau_P(q))$ points on one side of it. We can find h using different methods, e.g., by using the general reduction used by Aronov and Har-Peled [6] which can be summarized as follows: take 2^{-i}-random samples S_i of P, for $i = 1, \cdots, \log n$ until we find the first index l such that q lies outside the convex hull of S_l. Repeat this $O(\log n)$ times and let j be the smallest index found during these repetitions. Let h be the hyperplane that separates q from S_j. Testing whether q lies outside the convex hull of S_j and finding h if it does, is a three-dimensional linear programming step and can be done in $O(|S_j|)$ expected time so the overall computation time is $\tilde{O}(n)$. Let h^+ be the side of h that contains q. A standard application of Chernoff bound shows that with high probability, h^+ contains at least $\Omega(\tau_P(q)/\log n)$ points and at most $\Omega(\tau_P(q)\log n)$ points of P.

Similar to the technique used in Section 8, consider two parallel hyperplanes h_1 and h_2, but this time parallel to h. Do a central projection from q and map the points of P onto h_1 and h_2, resulting in point sets P_1 and P_2 on h_1 and h_2, respectively. However, we can observe that since h_1 and h_2 are parallel to h, one of them will contain $\tilde{O}(\tau_P(q))$ points. Let this be P_1.

Any simplex containing q must have at least one point from P_1. Thus, we can express the simplicial depth of q as the sum of the number of simplices containing exactly one point from P_1 (denoted by $\sigma^{(1)}(q)$) and the number of simplices containing two or more points from P_1 (denoted by $\sigma^{(2+)}(q)$). We approximate each term separately.

To approximate $\sigma^{(1)}(q)$, we build the data structure of Theorem 3.3 on P_2 in $\tilde{O}(n)$ time. For any point $p \in P_1$, consider the ray $\overrightarrow{pq}$ and its intersection p' with h_2. The two-dimensional simplicial depth $\sigma_{P_2}(p')$ approximates the number of triangles that contain p' which is equal to the number of three-dimensional simplices that contain q and have only p from P_1. Thus, by issuing $|P_1|$ queries to the two-dimensional data structure, we can approximate $\sigma^{(1)}(q)$ in $\tilde{O}(|P_1|) = \tilde{O}(n)$ time.

To approximate $\sigma_P^{(2+)}(q)$, notice that $\sigma_P^{(2+)}(q) \leq |P_1|^2 n^2$ since we are forced to pick at least two points from P_1. On the other hand, by the Lemma 2.2, $\sigma_P(q) = \Omega(\tau_P^3(q)n)$. Thus, we can use the same approach as in Subsection 6.2 and directly sample simplices. However, we sample simplices that have at least two points from P_2. Using the above inequalities and similar to the analysis used in Lemma 6.4, it suffices to sample $O(\frac{\delta \log n |P_1|^2 n^2}{\varepsilon^2 \tau_P^3(q) n}) = \tilde{O}(n)$ simplices and to obtain a relative $(1+\varepsilon)$-approximation with high probability for $\sigma_P^{(2+)}(q)$. $\square$

8 An Exact Algorithm in High Dimensions

In this section we describe a simple strategy to compute the simplicial depth exactly in $O(n^d \log n)$ time. While we do not achieve the conjectured lower bound of $\Omega(n^{d-1})$, we cut down roughly a factor n compared to the trivial upper bound of $O(n^{d+1})$. Note that this almost matches the best previous bound of $O(n^4)$ in 4D as well [21].

W.l.o.g, assume q is the origin, $\mathbf{0}$. Our main idea is very simple: consider d points $p_1, \ldots, p_d \in P$. Let $\overrightarrow{r_i}$ be the ray that originates from $\mathbf{0}$ towards $-p_i$. We would like to count how many points $p \in P$ can create a simplex with $p_1, \ldots, p_d$ that contains $\mathbf{0}$. We observe that this is equivalent to counting the number of points of P that lie inside the simplex created by rays $\overrightarrow{r_1}, \ldots, \overrightarrow{r_d}$. We can count this number in polylogarithmic time if we spend $\tilde{O}(n^d)$ time to build a simplex range counting data structure on P. This would give an algorithm with overall running time of $\tilde{O}(n^d)$. We can cut the log factors down to one by employing a slightly more intelligent approach.

We use the following observation made by Gil et al. [23].

Observation 8.1. *Let q be a point inside a simplex $a_1 \ldots a_{d+1}$ and let a_i' be a point on the ray $\overrightarrow{qa_i}$. The simplex defined by $a_1 \ldots a_{i-1} a_i' a_{i+1} \ldots a_{d+1}$ contains p.*

Pick two arbitrary parallel hyperplanes h_1 and h_2 such that P lies between them. This can be done easily in $O(n)$ time. Next, using central projection from $\mathbf{0}$, we map the points onto the hyperplanes h_1 and h_2: for every point $p_i \in P$, we create the ray $\overrightarrow{\mathbf{0}p_i}$ and let p_i' be the intersection of the ray with h_1 or h_2. Thus, the point set P can be mapped to two point sets P_1 and P_2

where P_1 lies on h_1 and p_2 lies on P_2 and furthermore, by Observation 8.1, $\sigma_P(q) = \sigma_{P_1 \cup P_2}(q)$.

Now we use the following result from the simplex range counting literature.

Theorem 8.2. *[20] Given a set of n points in d-dimensional space, and any constant $\varepsilon > 0$, one can build a data structure of size $O(n^{d+\varepsilon})$ in $O(n^{d+\varepsilon})$ expected preprocessing time, such that given any query simplex Δ, the number of points in Δ can be counted in $O(\log n)$ time.*

We build the above data structure on P_1 and P_2. However, since both of these point sets lie on a $(d-1)$-dimensional flat, the preprocessing time is $O(n^{d-1+\varepsilon}) = O(n^d)$ if we choose $\varepsilon = 1/2$. Next, for any d tuples of points $p_1, \dots, p_d$, we create the rays $\overrightarrow{r_1}, \dots, \overrightarrow{r_d}$ and the corresponding simplex Δ. We find the intersection of Δ in $O(1)$ time with hyperplanes h_1 and h_2 and issue two simplex range counting queries, one in each hyperplane. Thus, in $O(\log n)$ time, we can count how many simplices contain $\mathbf{0}$ that are made by points $p_1, \dots, p_d$. We add all these numbers over all d tuples, which counts each simplex containing $\mathbf{0}$ exactly $(d+1)$ times. The number of d-tuples is $O(n^d)$ and for each we spend $O(\log n)$ time querying the data structures. Thus, we obtain the following theorem.

Theorem 8.3. *Given a set P of n points in $\mathbb{R}^d$, the simplicial depth of a point p can be computed in $O(n^d \log n)$ expected time.*

9 Complexity

Let $P \subset \mathbb{R}^d$ be a set of n points and $q \in \mathbb{R}^d$ a query point. If the dimension is constant, then clearly computing $\sigma_P(q)$ can be carried out in polynomial time. We now consider the case that d is part of the input. We show that in this case computing the simplicial depth is #P-complete by a reduction from counting the number of perfect matchings in bipartite graphs.

Proposition 9.1. *Let $P \subset \mathbb{R}^d$ be a set and $q \in \mathbb{R}^d$ a query point. Then, computing $\sigma_P(q)$ is #P-complete if the dimension is part of the input.*

Proof. Let $G = (V, E)$ be a bipartite graph with $|V| = n$ and $|E| = m$. It is well known that computing the number of perfect matchings in G is #P-complete [32]. Let $\mathcal{P}_H \subset \mathbb{R}^m$ be the perfect matching polytope for G [24, Chapter 30]. It is defined by $m + 2n$ half-spaces. Furthermore, the number of vertices of $\mathcal{P}_H$ equals the number k of perfect matchings in G. Consider now the dual polytope $\mathcal{P}_V \subset \mathbb{R}^m$. It is the convex hull of $m + 2n$ points $P \subset \mathbb{R}^m$ and the number of facets equals k. Let $\bar{P} \subset \mathbb{R}^{2n-1}$ be the Gale transform of P. By Lemma 6.1, there is a bijection between the facets of $\mathcal{P}_V$ and the $(2n-1)$-simplices with vertices in $\bar{P}$ that contain $\mathbf{0}$ in their convex hull. Hence, $\sigma_{\bar{P}}(\mathbf{0}) = k$. $\square$

Next, we show that computing the simplicial depth is W[1]-hard with respect to the parameter d by a reduction to *d-Carathéodory*. In d-Carathéodory, we are given a set $P \subset \mathbb{R}^d$ and have to decide whether there is a $(d-1)$-simplex with vertices in P that contains $\mathbf{0}$ in its convex hull. Knauer et al. [27] proved that this problem is W[1]-hard with respect to the parameter d.

Proposition 9.2. *Let $P \subset \mathbb{R}^d$ be a set and $q \in \mathbb{R}^d$ a query point. Then, computing $\sigma_P(q)$ is W[1]-hard with respect to the parameter d.*

Proof. Assume we have access to an oracle that, given a query point q and a set $Q \subset \mathbb{R}^d$, returns $\sigma_Q(q)$. We show that #d-Carathéodory can be decided with two oracle queries.

Let k_d denote the number of $(d-1)$-simplices with vertices in P that contain $\mathbf{0}$ in their convex hulls and let k_{d+1} denote the number of d-simplices with vertices in P that contain $\mathbf{0}$ in their interior. Then $\sigma_P(\mathbf{0})$ can be written as $(|P|-d)k_d + k_{d+1}$. We want to decide whether $k_d > 0$. For each point $p \in P$ let $\tilde{p} \in \mathbb{R}^{d+1}$ denote the $(d+1)$-dimensional point that is obtained by appending a 1-coordinate and similarly, for each subset $P' \subset P$ let $\tilde{P'}$ denote the set $\{\tilde{p} \mid p \in P'\} \subset \mathbb{R}^{d+1}$. We denote with S the set $\{(0,\ldots,0,-1)^T, (0,\ldots,0,-2)^T\} \subset \mathbb{R}^{d+1}$ and set $Q = \tilde{P} \cup S$. Again, we want to express $\sigma_Q(\mathbf{0})$ as a function of k_d and k_{d+1}. Let $Q' \subset Q$, $|Q'| = d+2$, be a subset that contains $\mathbf{0}$ in its convex hull. Clearly, Q' has to contain a point from S. Let $\tilde{P'} = Q' \cap \tilde{P}$ denote the part from $\tilde{P}$ and let $S' = Q' \cap S$ denote the part from S. By construction of S, we have $(0,\ldots,0,1)^T \in \mathrm{conv}(\tilde{P'})$ and hence $\mathbf{0} \in \mathrm{conv}(P')$. That is, each $(d+2)$-simplex with vertices in Q that contains $\mathbf{0}$ in its convex hull corresponds to either a d-simplex or a $(d-1)$-simplex with vertices in P that contains $\mathbf{0}$ in its convex hull. Consider now a set $P' \subset P$ with $|P'| = d+1$ and $\mathbf{0} \in \mathrm{conv}(P)$. Then, the corresponding set $\tilde{P'}$ can be extended in two ways to a subset $Q' \subset Q$, $|Q'| = d+2$, with $\mathbf{0} \in \mathrm{conv}(Q')$ by taking either point in S. On the other hand, if $P' \subset P$ is a subset of size d with $\mathbf{0} \in \mathrm{conv}(P')$, then we can extend $\tilde{P'}$ to a set $Q' \subset Q$, $|Q'| = d+2$, with $\mathbf{0} \in \mathrm{conv}(Q')$ by either taking both points in S or by taking one arbitrary point in $\tilde{P} \setminus \tilde{P'}$ and either point in S. Hence, we have $\sigma_Q(\mathbf{0}) = 2k_{d+1} + k_{d-1} + 2(|P|-d)k_{d-1}$. Since $k_d = \sigma_Q(\mathbf{0}) - 2\sigma_P(\mathbf{0})$, we can decide whether $k_d > 0$ with two oracle queries. □

The following theorem is now immediate.

Theorem 9.3. *Let $P \subset \mathbb{R}^d$ be a set of d-dimensional points and $q \in \mathbb{R}^d$ a query point. Then, computing $\sigma_P(q)$ is #P-complete and W[1]-hard with respect to the parameter d.*

We conclude the section with constructive result: although computing the simplicial depth is #P-complete, it is possible to determine the parity in polynomial-time.

Proposition 9.4. *Let $P \subset \mathbb{R}^d$ be a set of points and $q \in \mathbb{R}^d$ a query point. If $n - d - 1$ is odd or $\binom{n}{d}$ is even, then $\sigma_P(q)$ is even. Otherwise, $\sigma_P(q)$ is odd.*

Proof. We assume w.l.o.g. that q is the origin. Since the simplicial graph $G_P(\mathbf{0})$ is $(n-d-1)$-regular, the product $(n-d-1)|V| = (n-d-1)\sigma_P(\mathbf{0})$ is even. If $(n-d-1)$ is odd, $\sigma_P(q)$ has to be even. Assume now $(n-d-1)$ is even. We construct a new point set Q in $\mathbb{R}^{d+1}$ similar as in the proof of Proposition 9.2. Let R denote the set $\{(0,\dots,0,-1)^T, (0,\dots,0,2)^T\} \subset \mathbb{R}^{d+1}$ and set $Q = \tilde{P} \cup R \subset \mathbb{R}^{d+1}$, where $\tilde{P}$ is defined as in the proof of Proposition 9.2. Let us now consider the graph $G_Q(\mathbf{0})$. Since $n-d-1$ is even, $(|Q| - (d+1) - 1) = n - d$ is odd. Now, $G_Q(\mathbf{0})$ is $(n-d)$-regular and thus $\sigma_Q(\mathbf{0})$ is even. Let $Q' \subset Q$, $|Q| = d+2$, be a subset that contains the origin in its convex hull. Then either (i) $R \subset Q'$ or (ii) Q' contains the point $r = (0,\dots,0,-1)^T \in R$ and $d+1$ points $\tilde{P'} \subseteq \tilde{P}$ with $(0,\dots,0,1)^T \in \operatorname{conv}(\tilde{P'})$. There are $\binom{n}{d}$ sets Q' with Property (i) and $\sigma_P(\mathbf{0})$ sets Q' with Property (ii). Hence, we have $\sigma_Q(\mathbf{0}) = \sigma_P(\mathbf{0}) + \binom{n}{d}$ is even and thus $\sigma_P(\mathbf{0})$ is odd iff $\binom{n}{d}$ is odd. □

Acknowledgements. This work was initiated while YS was visiting MADALGO in Aarhus. I would like to thank the working group for their hospitality and for a constructive atmosphere.

References

[1] Peyman Afshani, Lars Arge, and Kasper Green Larsen. Higher-dimensional orthogonal range reporting and rectangle stabbing in the pointer machine model. In *SCG '12: Symposium on Computational Geometry (SoCG)*, pages 323–332, 2012.

[2] Peyman Afshani and Timothy M. Chan. On approximate range counting and depth. *Discrete Comput. Geom.*, 42(1):3–21, 2009.

[3] Peyman Afshani, Chris Hamilton, and Norbert Zeh. A general approach for cache-oblivious range reporting and approximate range counting. *Computational Geometry: Theory and Applications*, 43:700–712, 2010.

[4] G. Aloupis, C. Cortés, F. Gómez, M. Soss, and G. Toussaint. Lower bounds for computing statistical depth. *Comput. Stat. & Data Anal.*, 40(2):223–229, 2002.

[5] Greg Aloupis. Geometric measures of data depth. *DIMACS series in discrete mathematics and theoretical computer science*, 72:147–158, 2006.

[6] B. Aronov and S. Har-Peled. On approximating the depth and related problems. *SIAM J. Comput.*, 38(3):899–921, 2008.

[7] B. Aronov, S. Har-Peled, and M. Sharir. On approximate halfspace range counting and relative epsilon-approximations. In *SCG '07: Symposium on Computational Geometry (SoCG)*, pages 327–336, 2007.

[8] Amitabha Bagchi, Amitabh Chaudhary, David Eppstein, and Michael T Goodrich. Deterministic sampling and range counting in geometric data streams. *ACM Transactions on Algorithms (TALG)*, 3(2):16, 2007.

[9] I. Bárány. A generalization of Carathéodory's theorem. *Discrete Math.*, 40(2):141–152, 1982.

[10] Jon Louis Bentley. Multidimensional divide-and-conquer. *Communications of the ACM*, 23(4):214–229, 1980.

[11] Michael Burr, Eynat Rafalin, and Diane L. Souvaine. Simplicial depth: An improved definition, analysis, and efficiency for the finite sample case. ftp://dimacs.rutgers.edu/pub/dimacs/TechnicalReports/TechReports/2003/2003-28.ps.gz.

[12] Michael Burr, Eynat Rafalin, and Diane L. Souvaine. Simplicial depth: An improved definition, analysis, and efficiency for the finite sample case. In *CCCG*, pages 136–139, 2004.

[13] Timothy M. Chan. Random sampling, halfspace range reporting, and construction of ($\leq k$)-levels in three dimensions. *SIAM J. Comput.*, 30(2):561–575, 2000.

[14] Timothy M. Chan, Kasper Green Larsen, and Mihai Pătraşcu. Orthogonal range searching on the RAM, revisited. In *SCG '11: Symposium on Computational Geometry (SoCG)*, pages 1–10, 2011.

[15] Timothy M Chan and Konstantinos Tsakalidas. Optimal deterministic algorithms for 2-d and 3-d shallow cuttings. In *SCG '15: Symposium on Computational Geometry (SoCG)*, 2015.

[16] B. Chazelle. *The discrepancy method: randomness and complexity*. Cambridge University Press, 2000.

[17] Bernard Chazelle. Filtering search: a new approach to query answering. *SIAM Journal of Computing*, 15(3):703–724, 1986.

[18] Bernard Chazelle and Leonidas J. Guibas. Fractional cascading: II. applications. *Algorithmica*, 1:163–191, 1986.

[19] Bernard Chazelle, Leonidas J. Guibas, and D. T. Lee. The power of geometric duality. *BIT Numerical Mathematics*, 25(1):76–90, 1985.

[20] Bernard Chazelle, Micha Sharir, and Emo Welzl. Quasi-optimal upper bounds for simplex range searching and new zone theorems. *Algorithmica*, 8:407–429, December 1992.

[21] A. Y. Cheng and M. Ouyang. On algorithms for simplicial depth. In *CCCG*, pages 53–56, 2001.

[22] Jeff Erickson. New lower bounds for convex hull problems in odd dimensions. *SIAM Journal on Computing*, 28(4):1198–1214, 1999.

[23] J. Gil, W. Steiger, and A. Wigderson. Geometric medians. *Discrete Math.*, 108(1):37–51, 1992.

[24] R. L. Graham, M. Grötschel, and L. Lovász. *Handbook of combinatorics*, volume 2. Elsevier, 1995.

[25] H. Kaplan and M. Sharir. Randomized incremental constructions of three-dimensional convex hulls and planar Voronoi diagrams, and approximate range counting. In *Proc. of the 17th Annu. ACM-SIAM Symp. on Discrete Algorithms*, pages 484–493, 2006.

[26] S. Khuller and J. S. B. Mitchell. On a triangle counting problem. *Inf. Process. Lett.*, 33(6):319–321, 1990.

[27] C. Knauer, H. R. Tiwary, and D. Werner. On the computational complexity of Ham-Sandwich cuts, Helly sets, and related problems. In *28th Int. Symp. on Theoretical Aspects of Comput. Sci. (STACS 2011)*, pages 649–660, 2011.

[28] R. Thomas R. *Lectures in Geometric Combinatorics.* American Mathematical Society, 2006.

[29] E. A. Ramos. On range reporting, ray shooting and k-level construction. In *SCG '99: Symposium on Computational Geometry (SoCG)*, pages 390–399. ACM, 1999.

[30] Jiří Matoušek. Reporting points in halfspaces. *Computational Geometry, Theory and Applications*, 2(3):169–186, 1992.

[31] P. J. Rousseeuw and I. Ruts. Bivariate location depth. *J. R. Stat. Soc. Series C (Appl. Stat.)*, 45(4):516–526, 1996.

[32] L. G. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8(2):189 – 201, 1979.