

# How to Get Close to the Median Shape\*

Sariel Har-Peled<sup>†</sup>

November 27, 2005

*They sought it with thimbles, they sought it with care;  
They pursued it with forks and hope;  
They threatened its life with a railway-share;  
They charmed it with smiles and soap.  
– The Hunting of the Snark, Lewis Carol*

## Abstract

In this paper, we study the problem of  $L_1$ -fitting a shape to a set of point, where the target is to minimize the sum of distances of the points to the shape, or alternatively the sum of squared distances. We present a general technique for computing a  $(1 + \varepsilon)$ -approximation for such a problem, with running time  $O(n + \text{poly}(\log n, 1/\varepsilon))$ , where  $\text{poly}(\log n, 1/\varepsilon)$  is polynomial of constant degree of  $\log n$  and  $1/\varepsilon$ . This is a linear time algorithm for a fixed  $\varepsilon > 0$ , and it is the first subquadratic algorithm for this problem.

Applications of the algorithm include best fitting either a circle, a sphere or a cylinder to a set of points when minimizing the sum of distances (or squared distances) to the respective shape.

## 1 Introduction

Consider the problem of fitting a parameterized shape to given data. This is a natural problem that arises in statistics, learning, data-mining and many other fields. What measure is being used for the quality of fitting has considerable impact of the hardness of the problem of finding the best fitting shape. As a concrete example, let  $P$  be a set of  $n$  points in  $\mathbb{R}^d$ . A typical criterion for measuring how well a shape  $\gamma$  fits  $P$ , denoted as  $\mu(P, \gamma)$ , is the maximum distance between a point of  $P$  and its nearest point on  $\gamma$ , i.e.,  $\mu(P, \gamma) = \max_{p \in P} d(p, \gamma)$ , where  $d(p, \gamma) = \min_{q \in \gamma} \|p - q\|$ . The extent measure of  $P$  is  $\mu(P) = \min_{\gamma \in \mathcal{F}} \mu(P, \gamma)$ , where  $\mathcal{F}$  is a family of shapes (such as points, lines, hyperplanes, spheres, etc). For example, the problem of finding the minimum radius sphere (resp. cylinder) enclosing  $P$  is the same as finding the point (resp. line) that fits  $P$  best, and the problem of finding the smallest width slab

---

\*Alternative titles for this paper include: “How to stay connected with your inner circle” and “How to compute one ring to rule them all”.

<sup>†</sup>Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA; [sariel@uiuc.edu](mailto:sariel@uiuc.edu); <http://www.uiuc.edu/~sariel/>. Work on this paper was partially supported by a NSF CAREER award CCR-0132901.

(resp. spherical shell, cylindrical shell) is the same as finding the hyperplane (resp. sphere, cylinder) that fits  $P$  best.

A natural way of encoding the fitting information for a given a shape  $\gamma$  is to represent the distances of a shape  $\gamma$  to  $P$ , by creating a point  $\mathbf{d}(P, \gamma) \in \mathbb{R}^n$ , where the  $i$ th coordinate is the distance of the  $i$ th point of  $P$  from  $\gamma$ . Thus, the shape fitting problem mentioned above (of minimizing the distance to the furthest point to the shape), asks to minimize the shape  $\gamma$  that realizes  $\min_{\gamma \in \mathcal{F}} \|\mathbf{d}(P, \gamma)\|_\infty$ . We will refer to this as the  $L_\infty$ -shape fitting problem.

The exact algorithms for best shape fitting are generally expensive, e.g., the best known algorithms for computing the smallest volume bounding box containing  $P$  in  $\mathbb{R}^3$  require  $O(n^3)$  time. Consequently, attention has shifted to developing approximation algorithms [BH01, ZS02]. A general approximation technique was recently developed for such problems by Agarwal *et al.* [AHV04]. This implies among other things that one can approximate the circle that best fit a set of points in the plane in  $O(n + 1/\varepsilon^{O(1)})$  time, where the fitting measure is the maximum distance of the point to the circle (in fact, this special case was handled before by Agarwal *et al.* [AAHS00] and by Chan [Cha02]).

The main problem with the  $L_\infty$ -fitting, is its sensitivity to noise and outliers. There are two natural remedies.

The first is to change the target function to be less sensitive to outliers. For example, instead of considering the maximum distance, one can consider the average distance. This is the  $L_1$ -fitting problem, and here we would like to compute the shape realizing  $\ell_1(\mathcal{F}, P) = \min_{\gamma \in \mathcal{F}} \|\mathbf{d}(P, \gamma)\|_1 = \min_{\gamma \in \mathcal{F}} \sum_{p \in P} d(p, \gamma)$ . Similarly, in the  $L_2$ -fitting problem, one would like to minimize the average squared distances of the points to the shape; namely,  $\ell_2(\mathcal{F}, P) = \min_{\gamma \in \mathcal{F}} \|\mathbf{d}(P, \gamma)\|_2^2 = \min_{\gamma \in \mathcal{F}} \sum_{p \in P} (d(p, \gamma))^2$ . The  $L_2$  fitting problem in the case of a single linear subspace is well understood, and is computed via SVD (singular value decomposition). Fast approximation algorithms are known for this problem, see [FKV98, RVW04] and references therein. As for the  $L_1$ -fitting of a linear subspace, this problem can be solved using linear programming techniques, in polynomial time in high dimensions, and linear time in constant dimension [YKII88]. Recently, Clarkson gave a faster approximation algorithm for this problem [Cla05] which works via sampling.

The problem seems to be harder once the shape we consider is not a linear subspace. There is considerable work on nonlinear regressions [SW89] (i.e., extension of the  $L_2$  least squares technique) for various shapes, but there not seems to be an efficient guaranteed approximation algorithm even for the “easy” problem of  $L_1$ -fitting a circle to the data. The hardness seems to rise from the the target function being a sum of terms, each term being an absolute value of a difference of a square root of a polynomial and a radius (see Section 2.1.2). In fact, this is an extension of the Fermat-Weber problem and it seems doubtful that efficient exact solution would exist for such a problem.

The second approach is to specify a number  $k$  of outliers in advance and find the best shape  $L_\infty$ -fitting all but  $k$  of the input points. Har-Peled and Wang showed that there is a coresets for this problem [HW04], and as such it can be solved in  $O(n + \text{poly}(k, \log n, 1/\varepsilon))$  time, for a large family of shapes. The work of Har-Peled and Wang was motivated by the aforementioned problem of  $L_1$ -fitting a circle to a set of points. (The results of Har-Peled and Wang were recently improved by Agarwal *et al.* [AHY06], but since the improvement is

not significant for our purposes we will stick with the older reference.)

**Our Results.** In this paper, we describe a general technique for computing  $(1 + \varepsilon)$ -approximate solution to the  $L_1$  and  $L_2$ -fitting problems, for a family of shapes which is well behaved (roughly speaking, those are all the shapes that the technique of Agarwal *et al.* [AHV04] can handle). Our algorithm achieves a running time of  $O(n + \text{poly}(\log n, 1/\varepsilon))$ . As such, this work can be viewed as the counterpart to Agarwal *et al.* [AHV04] work on the approximate  $L_\infty$ -fitting problem. This is the first linear time algorithm for this problem.

The only previous algorithm directly relevant for this result, we are aware of, is due to Har-Peled and Koltun [HK04a] which in  $O(n^2\varepsilon^{-2}\log^2 n)$  time approximates the best circle  $L_1$ -fitting a set of points in the plane.

The paper is organized as follows. In Section 2 we introduce some necessary preliminaries. In Section 2.1 the problem is stated formally. In Section 3, we provide a (somewhat bizarre) solution for the one-point  $L_1$ -fitting problem in one dimension (i.e., the one median problem in one dimension). In Section 4, we show how the problem size can be dramatically reduced. In Section 5, a slow approximation algorithm is described for the problem (similar in nature to the algorithm of [HK04a]). In Section 6, we state our main result and some applications. Conclusions are provided in Section 7.

## 2 Preliminaries

Throughout the paper, we refer to the  $x_d$ -parallel direction in  $\mathbb{R}^d$  as *vertical*. Given a point  $x = (x_1, \dots, x_{d-1})$  in  $\mathbb{R}^{d-1}$ , let  $(x, x_d)$  denote the point  $(x_1, \dots, x_{d-1}, x_d)$  in  $\mathbb{R}^d$ . Each point  $x \in \mathbb{R}^d$  is also a vector in  $\mathbb{R}^d$ . Given a geometric object  $A$ ,  $A + x$  represents the object obtained by translating each point in  $A$  by  $x$ .

A *surface* is a subset of  $\mathbb{R}^d$  that intersects any vertical line in a single point. A *surface patch* is a portion of a surface such that its vertical projection into  $\mathbb{R}^{d-1}$  is a semi-algebraic set of constant complexity usually a simplex. Let  $A$  and  $B$  be either a point, a hyperplane, or a surface in  $\mathbb{R}^d$ . We say that  $A$  lie *above* (resp. *below*)  $B$ , denoted by  $A \succeq B$  (resp.  $A \preceq B$ ), if for any vertical line  $\ell$  intersecting both  $A$  and  $B$ , we have that  $x_d \geq y_d$  (resp.  $x_d \leq y_d$ ), where  $(x_1, \dots, x_{d-1}, x_d) = A \cap \ell$  and  $(x_1, \dots, x_{d-1}, y_d) = B \cap \ell$ .

Two non-negative numbers  $x$  and  $y$  are  $\varepsilon$ -*approximation* of each other if  $(1 - \varepsilon)x \leq y \leq (1 + \varepsilon)x$  and  $(1 - \varepsilon)y \leq x \leq (1 + \varepsilon)y$ . We denote this fact by  $x \approx_\varepsilon y$ . Two non-negative functions  $f(\cdot)$  and  $g(\cdot)$  are  $\varepsilon$ -*approximation* of each other, denoted by  $f \approx_\varepsilon g$ , if  $f(x) \approx_\varepsilon g(x)$ , for all  $x$ .

**Observation 2.1** *Let  $x$  and  $y$  be two positive numbers and  $\varepsilon < 1/4$ . We have: (i) If  $x \approx_\varepsilon y$  and  $y \approx_\varepsilon z$  then  $x \approx_{3\varepsilon} z$ . (ii) If  $|x - y| \leq \varepsilon x$  then  $x \approx_{2\varepsilon} y$ . (iii) If  $x \leq (1 + \varepsilon)y$  and  $y \leq (1 + \varepsilon)x$  then  $x \approx_\varepsilon y$ .*

## 2.1 Problem Statement

### 2.1.1 The Circle Fitting Case

To motivate our exposition we will first consider the problem of  $L_1$ -fitting a circle to a set of points in the plane.

Let  $P = \{p_1, \dots, p_n\}$  be a set of  $n$  points in the plane, and consider the price  $\nu_P(C)$  of  $L_1$ -fitting the circle  $C$  to  $P$ . Formally, for a point  $p_i \in P$  let  $f_i(C) = \left| \|p_i - c\| - r \right|$ , where  $c$  is the center of  $C$ , and  $r$  is the radius of  $C$ . Thus, the overall price, for a circle  $C$  centered at  $(x, y)$  with radius  $r$ , is

$$\nu_P(C) = \nu_P(x, y, r) = \sum_{i=1}^n f_i(C) = \sum_{i=1}^n \left| \|p_i - c\| - r \right| = \sum_{i=1}^n \left| \sqrt{(x_i - x)^2 + (y_i - y)^2} - r \right|,$$

where  $p_i = (x_i, y_i)$ , for  $i = 1, \dots, n$ . We are looking for the circle  $C$  minimizing  $\nu_P(C)$ . This is the circle that best fits the point set under the  $L_1$  metric. Let  $\nu_{\text{opt}}(P)$  denote the price of the optimal circle  $C_{\text{opt}}$ .

Geometrically, each function  $f_i$  induces a surface  $\gamma_i = \cup_{p \in \mathbb{R}^2} (x_p, y_p, \|p - p_i\|)$  in 3D, which is a cone. A circle is encoded by a point  $C = (x, y, r)$ . The value of  $f_i(C)$  is the vertical distance between the point  $C$  and surface  $\gamma_i$ . Thus, we have a set  $\mathcal{G}$  of  $n$  surfaces in 3D, and we are interested in finding the point that minimizes the sum of vertical distances of this point to the  $n$  surfaces.

### 2.1.2 The General Problem

Formally, for a weighted set of surfaces  $\mathcal{G}$  in  $\mathbb{R}^d$  and  $p$  any point in  $\mathbb{R}^d$  let  $\nu_{\mathcal{G}}(p) = \sum_{\gamma \in \mathcal{G}} w_{\gamma} \cdot d_{\perp}(p, \gamma)$  denote the  $L_1$  distance of  $p$  from  $\mathcal{G}$ , where  $d_{\perp}(p, \gamma)$  is the vertical distance between  $p$  and the surface  $\gamma$  and  $w_{\gamma}$  is the weight associated with  $\gamma$ . Throughout our discussion weights are positive integer numbers. If  $\mathcal{G}$  is unweighted then any surface  $\gamma \in \mathcal{G}$  is assigned weight  $w_{\gamma} = 1$ . We would be interested in finding the point that minimizes  $\nu_{\mathcal{G}}(p)$  when  $p$  is restricted to a domain  $\mathcal{D}_d$ , which is a semi-algebraic set of constant complexity. This is the  $L_1$ -fitting problem. The  $L_2$ -fitting problem is computing the point  $p \in \mathcal{D}_d$  realizing the minimum of  $\mu_{\mathcal{G}}(p) = \sum_{\gamma \in \mathcal{G}} w_{\gamma} \cdot (d_{\perp}(p, \gamma))^2$ .

It would be sometime conceptually easier to think about the problem algebraically, where the  $i$ th surface  $\gamma_i$  is an image of a (non-negative)  $(d-1)$ -dimensional function  $f_i(x_1, \dots, x_{d-1}) = \sqrt{p_i(x_1, \dots, x_{d-1})}$ , where  $p_i(\cdot)$  is a constant degree polynomial, for  $i = 1, \dots, n$ . We are interested in approximating one of the following quantities:

$$\begin{aligned} \text{(i)} \quad & \min_{(x_1, \dots, x_{d-1}) \in \mathcal{D}} \sum_{i=1}^n w_i \cdot f_i(x_1, \dots, x_{d-1}), \\ \text{(ii)} \quad & \nu_{\text{opt}}(\mathcal{G}) = \min_{x \in \mathcal{D}_d} \nu_{\mathcal{G}}(x) = \min_{(x_1, \dots, x_d) \in \mathcal{D}_d} \sum_i w_i \cdot |f_i(x_1, \dots, x_{d-1}) - x_d|, \\ \text{or (iii)} \quad & \mu_{\text{opt}}(\mathcal{G}) = \min_{x \in \mathcal{D}_d} \mu_{\mathcal{G}}(x) = \min_{(x_1, \dots, x_d) \in \mathcal{D}_d} \sum_i w_i \cdot (f_i(x_1, \dots, x_{d-1}) - x_d)^2, \end{aligned}$$

where  $\mathcal{D}$  and  $\mathcal{D}_d$  are semi-algebraic sets of constant complexity, and the weights  $w_1, \dots, w_n$  are positive integers. Note, that (i) is a special case of (ii), by setting  $\mathcal{D}_d = \mathcal{D} \times \{0\}$ .

To simplify the exposition, we will assume that  $\mathcal{D}_d = \mathbb{R}^d$ . It is easy to verify that our algorithm works also for the more general case with a few minor modifications.

**The linearization dimension.** In the following, a significant parameter in the exposition is the *linearization dimension*  $\mathbf{s}$ , which is the target dimension we need to map the polynomials  $p_1, \dots, p_n$  so that they all become linear functions. For example, if the polynomials are of the form  $p_i(x, y, z) = x^2 + y^2 + z^2 + a_i x + b_i y + c_i z$ , for  $i = 1, \dots, n$ , then they can be linearized by a mapping  $\mathbf{L}(x, y, z) = (x^2 + y^2 + z^2, x, y, z)$ , such that  $h_i(x, y, z, w) = w + a_i x + b_i y + c_i z$  is a linear function and  $g_i(x, y, z) = h_i(\mathbf{L}(x, y, z))$ . The linearization dimension is always bounded by the number of different monomials appearing in the polynomials  $p_1, \dots, p_n$ . Agarwal and Matoušek [AM94] describe an algorithm that computes a linearization of the smallest dimension for a family of such polynomials.

### 3 Approximate $L_1$ -Fitting in One Dimension

In this section, we consider the one dimensional problem of approximating the distance function of a point  $z$  to a set of points  $\mathbf{Z} = \langle z_1, z_2, \dots, z_n \rangle$ , where  $z_1 \leq z_2 \leq \dots \leq z_n$ . Formally, we want to approximate the function  $\nu_{\mathbf{Z}}(\mathbf{z}) = \sum_{z_i \in \mathbf{Z}} |z_i - \mathbf{z}|$ . This is the one median function for  $\mathbf{Z}$  on the real line. This corresponds to a vertical line in  $\mathbb{R}^d$ , where each  $z_i$  represents the intersection of the vertical line with the surface  $\gamma_i$ . The one dimensional problem is well understood and there exists a coresets for it, see [HM04, HK04b]. Unfortunately, it is unclear how to extend these constructions to the higher dimensional case; specifically, how to perform the operations required in a global fashion on the surfaces so that the construction would hold for all vertical lines. Thus, we present here an alternative construction.

**Definition 3.1** For a set of surfaces  $\mathcal{G}$  in  $\mathbb{R}^d$ , a weighted subset  $\mathcal{S} \subseteq \mathcal{G}$  is  $\varepsilon$ -coreset for  $\mathcal{G}$  if for any point  $p \in \mathbb{R}^d$  we have  $\nu_{\mathcal{G}}(p) \approx_{\varepsilon} \nu_{\mathcal{S}}(p)$ .

The first step is to partition the points. Formally, we partition  $\mathbf{Z}$  symmetrically into subsets, such that the size of the subsets increase in size as one comes toward the middle of the set. Formally, the set  $L_i = \{z_i\}$  contains the  $i$ th point on the line, for  $i = 1, \dots, \mathbf{m}$ , where  $\mathbf{m} \geq 10/\varepsilon$  is a parameter to be determined shortly. Similarly,  $R_i = \{z_{n-i+1}\}$ , for  $i = 1, \dots, \mathbf{m}$ . Set  $\alpha_{\mathbf{m}} = \mathbf{m}$ , and let  $\alpha_{i+1} = \min(\lceil (1 + \varepsilon/10)\alpha_i \rceil, n/2)$ , for  $i = \mathbf{m}, \dots, \mathbf{M}$ , where  $\alpha_{\mathbf{M}}$  is the first number in this sequence equal to  $n/2$ . Now, let  $L_i = \{z_{\alpha_{i-1}+1}, \dots, z_{\alpha_i}\}$  and  $R_i = \{z_{n-\alpha_{i-1}}, \dots, z_{n-\alpha_i+1}\}$ , for  $i = \mathbf{m} + 1, \dots, \mathbf{M}$ . We will refer to a set  $L_i$  or  $R_i$  as a *chunk*. Consider the partition of  $\mathbf{Z}$  formed by the chunks  $L_1, L_2, \dots, L_{\mathbf{M}}, R_{\mathbf{M}}, \dots, R_2, R_1$ . Clearly, this is a partition of  $\mathbf{Z}$  into “exponential sets”. The first/last  $\mathbf{m}$  sets on the boundary are singletons, and all the other sets grow exponentially in cardinality, till they cover the whole set  $\mathbf{Z}$ .

Next, we pick an arbitrary points  $l_i \in L_i$  and  $r_i \in R_i$  and assign them weight  $w_i = |R_i| = |L_i|$ , for  $i = 1, \dots, \mathbf{M}$ . Let  $\mathcal{S}$  be the resulting weighted set of points. We claim that this is a coresets for the 1-median function.

But before delving into this, we need the following technical lemma.

**Lemma 3.2** *Let  $A$  be a set of  $n$  real numbers, and let  $\psi$  and  $\mathbf{z}$  be any two real numbers. We have that  $\left| \nu_A(\mathbf{z}) - |A| \cdot |\psi - \mathbf{z}| \right| \leq \nu_A(\psi)$ .*

*Proof:* Omitted. Included in the online full-version of this paper [Har05]. ■

**Lemma 3.3** *It holds  $\nu_{\mathbf{Z}}(\mathbf{z}) \approx_{\varepsilon/5} \nu_{\mathcal{S}}(\mathbf{z})$ , for any  $\mathbf{z} \in \mathbb{R}$ .*

*Proof:* We claim that

$$\left| \nu_{\mathbf{Z}}(\mathbf{z}) - \nu_{\mathcal{S}}(\mathbf{z}) \right| \leq (\varepsilon/10) \nu_{\mathbf{Z}}(\mathbf{z}),$$

for all  $\mathbf{z} \in \mathbb{R}$ . Indeed, let  $\tau$  be a median point of  $\mathbf{Z}$  and observe that  $\nu_{\mathbf{Z}}(\tau)$  is a global minimum of this function. We have that

$$\begin{aligned} \mathcal{E} = \left| \nu_{\mathbf{Z}}(\mathbf{z}) - \nu_{\mathcal{S}}(\mathbf{z}) \right| &\leq \sum_{i=1}^M \left| \nu_{L_i}(\mathbf{z}) - |L_i| \cdot |l_i - \mathbf{z}| \right| + \sum_{i=1}^M \left| \nu_{R_i}(\mathbf{z}) - |R_i| \cdot |r_i - \mathbf{z}| \right| \\ &= \sum_{i=m+1}^M \left| \nu_{L_i}(\mathbf{z}) - |L_i| \cdot |l_i - \mathbf{z}| \right| + \sum_{i=m+1}^M \left| \nu_{R_i}(\mathbf{z}) - |R_i| \cdot |r_i - \mathbf{z}| \right| \\ &\leq \sum_{i=m+1}^M \nu_{L_i}(l_i) + \sum_{i=m+1}^M \nu_{R_i}(r_i), \end{aligned}$$

by Lemma 3.2.

Observe that by construction  $|R_i| \leq (\varepsilon/10) |R_1 \cup \dots \cup R_{i-1}|$ , for  $i > m$ . We claim that this implies that  $\sum_{i=m+1}^M \nu_{L_i}(l_i) + \sum_{i=m+1}^M \nu_{R_i}(r_i) \leq (\varepsilon/10) \nu_{\mathbf{Z}}(\tau)$ . To see that, for each point of  $z_i \in \mathbf{Z}$ , let  $I_i$  be the interval with  $z_i$  in one endpoint, and the median  $\tau$  as the other endpoint. The total length of those intervals is  $\nu_{\mathbf{Z}}(\tau)$ . Let  $\mathfrak{K} = \{I_1, \dots, I_n\}$ .

Consider the interval  $\mathcal{I}_i = \mathcal{I}(R_i)$  which is the shortest interval containing the points of  $R_i$ , for  $i = m+1, \dots, M$ . Clearly, we have  $\nu_{R_i}(r_i) \leq |R_i| \cdot \|\mathcal{I}_i\|$ .

On the other hand, the number of intervals of  $\mathfrak{K}$  completely covering  $\mathcal{I}_i$  is at least  $(10/\varepsilon) |R_i|$ , for  $i = m+1, \dots, M$ . As such, we can charge the total length of  $\nu_{R_i}(r_i)$  to the portions of those intervals of  $\mathfrak{K}$  covering  $\mathcal{I}_i$ . Thus, every unit of length of the intervals of  $\mathfrak{K}$  get charged at most  $\varepsilon/10$  units.

This implies that the error  $\mathcal{E} \leq (\varepsilon/10) \nu_{\mathbf{Z}}(\tau) \leq (\varepsilon/10) \nu_{\mathbf{Z}}(\mathbf{z})$ , which establishes the lemma, by Observation 2.1. ■

Next, we “slightly” perturb the points of the coreset  $\mathcal{S}$ . Formally, assume that we have points  $l'_1, \dots, l'_M, r'_1, \dots, r'_M$  such that  $|l'_i - l_i|, |r'_i - r_i| \leq (\varepsilon/20) |l_i - r_i|$ , for  $i = 1, \dots, N$ . Let  $\mathcal{R} = \{l'_1, \dots, l'_M, r'_1, \dots, r'_M\}$  be the resulting weighted set. We claim that  $\mathcal{R}$  is still a good coreset.

**Lemma 3.4** *It holds that  $\nu_{\mathbf{Z}}(\mathbf{z}) \approx_{\varepsilon} \nu_{\mathcal{R}}(\mathbf{z})$ , for any  $\mathbf{z} \in \mathbb{R}$ . Namely,  $\mathcal{R}$  is a  $\varepsilon$ -coreset for  $\mathbf{Z}$ .*

*Proof:* By Lemma 3.3 and by the triangle inequality, we have

$$\nu_{\mathcal{S}}(\mathbf{z}) = \sum_i (|L_i| \cdot |l_i - \mathbf{z}| + |R_i| \cdot |r_i - \mathbf{z}|) \geq \sum_i |L_i| \cdot |l_i - r_i|,$$

since for all  $i$  we have  $|L_i| = |R_i|$ . Also, by the triangle inequality  $||l_i - \mathbf{z}| - |l'_i - \mathbf{z}|| \leq |l_i - l'_i|$ . Thus

$$\left| \nu_{\mathcal{S}}(\mathbf{z}) - \nu_{\mathcal{R}}(\mathbf{z}) \right| \leq \sum_i |L_i| \cdot |l_i - l'_i| + \sum_i |R_i| |r_i - r'_i| \leq 2 \sum_i |L_i| \frac{\varepsilon}{20} \cdot |l_i - r_i| \leq \frac{\varepsilon}{10} \nu_{\mathcal{S}}(\mathbf{z}).$$

Thus  $\mathcal{R}$  is a  $\varepsilon/5$ -coreset of  $\mathcal{S}$ , which is in turn a  $\varepsilon$ -coreset for  $\mathbf{Z}$ , by Observation 2.1. ■

### 3.1 Variants

Let  $f : \mathbb{R}^+ \rightarrow \mathbb{R}^+$  be a monotone strictly increasing function (i.e.,  $f(x) = x^2$ ). Consider the function

$$U_P(\mathbf{z}) = \sum_{x \in P} f(|x - \mathbf{z}|).$$

We claim that the set  $\mathcal{S}$  constructed in Lemma 3.3 is also a coreset for  $U_P(\cdot)$ . Namely,  $U_P(\mathbf{z}) \underset{\varepsilon/5}{\approx} U_{\mathcal{S}}(\mathbf{z}) = \sum_{x \in \mathcal{S}} w_x f(|x - \mathbf{z}|)$ . To this end, map each point  $x$  of  $P$ , to a point of distance  $f(|x - \mathbf{z}|)$  from  $\mathbf{z}$  (preserving the side of  $\mathbf{z}$  on which the point  $x$  lies), and let  $g_{\mathbf{z}} : P \rightarrow \mathbb{R}$  denote this mapping. Let the resulting set be  $Q = f(P)$ . Clearly,  $U_P(\mathbf{z}) = \nu_Q(\mathbf{z})$ , and let  $\mathcal{T}$  be the coreset constructed for  $Q$  by the construction of Lemma 3.3. Observe that  $\mathcal{T} = g_{\mathbf{z}}(\mathcal{S})$ , since the construction of the coreset cares only about the ordering of the points, and the ordering is preserved when mapping between  $P$  and  $Q$ . Thus, we have that  $U_P(\mathbf{z}) = \nu_Q(\mathbf{z}) \underset{\varepsilon/5}{\approx} \nu_{\mathcal{T}}(\mathbf{z}) = U_{\mathcal{S}}(\mathbf{z})$ , as required.

This in particular implies that  $\mu_P(\mathbf{z}) \underset{\varepsilon/5}{\approx} \mu_{\mathcal{S}}(\mathbf{z})$ , for any  $\mathbf{z} \in \mathbb{R}$ , where  $\mu_P(\mathbf{z}) = \sum_{x \in P} |\mathbf{z} - x|^2$ . In this case, even the modified coreset  $\mathcal{R}$  is still a coreset.

**Lemma 3.5** *It holds that  $\mu_{\mathbf{Z}}(\mathbf{z}) \underset{\varepsilon}{\approx} \mu_{\mathcal{R}}(\mathbf{z})$ , for any  $\mathbf{z} \in \mathbb{R}$ . Namely,  $\mathcal{R}$  is a  $\varepsilon$ -coreset of  $\mathbf{Z}$  for the  $\mu(\cdot)$  function.*

*Proof:* Omitted. Included in the online full-version of this paper [Har05]. ■

## 4 The Reduction

In this section, we show how to reduce the problem of approximating the  $\nu_{\mathcal{G}}(\cdot)$  function, for a set  $\mathcal{G}$  of  $n$  surfaces in  $\mathbb{R}^d$ , to the problem of approximating the same function for a considerably smaller set of surface patches.

Section 3 provides us with a general framework for how to get a small approximation. Indeed, pick any vertical line  $\ell$ , and consider its intersection points with the surfaces of  $\mathcal{G}$ . Clearly, the function  $\nu_{\mathcal{G}}(\cdot)$  restricted to  $\ell$  can be approximated using the construction of Section 3. To this end, we need to pick levels in the way specified and assign them the appropriate weights. This would guarantee that the resulting function would approximate  $\nu_{\mathcal{G}}(\cdot)$  everywhere.

A major difficulty in perusing this direction is that the levels we pick have high descriptive complexity. We circumnavigate this difficulty in two stages. In the first stage, we replace those levels by shallow levels, by using random sampling. In the second stage, we approximate these shallow levels such that this introduces small relative error.

**Definition 4.1** For a set  $\mathcal{G}$  of  $n$  surfaces in  $\mathbb{R}^d$ , the *level* of a point  $x \in \mathbb{R}^d$  in the arrangement  $\mathcal{A}(\mathcal{G})$  is the number of surfaces of  $\mathcal{G}$  lying vertically below  $x$ . For  $k = 0, \dots, n-1$ , let  $\mathbf{L}_{\mathcal{G},k}$  represent the surface which is closure of all points on the surfaces of  $\mathcal{G}$  whose level is  $k$ . We define the *top  $k$ -level* of  $\mathcal{G}$  to be  $\mathbf{U}_{\mathcal{G},k} = \mathbf{L}_{\mathcal{G},n-k-1}$ , for  $k = 0, \dots, n-1$ . Note that  $\mathbf{L}_{\mathcal{G},k}$  is a subset of the arrangement of  $\mathcal{G}$ . For  $x \in \mathbb{R}^{d-1}$ , we slightly abuse notations and define  $\mathbf{L}_{\mathcal{G},k}(x)$  to be the value  $x_d$  such that  $(x, x_d) \in \mathbf{L}_{\mathcal{G},k}$ .

**Lemma 4.2** Let  $\mathcal{G}$  be a set of  $n$  surfaces in  $\mathbb{R}^d$ ,  $0 < \delta < 1/4$ , and let  $k$  be a number between 0 and  $n/2$ . Let  $\zeta = \min(ck^{-1}\delta^{-2} \log n, 1)$ , and pick each surface of  $\mathcal{G}$  into a random sample  $\mathcal{R}$  with probability  $\zeta$ , where  $c$  is an appropriate constant. Then, with high probability, the  $\tilde{\kappa} = \zeta k = O(\delta^{-2} \log n)$  level of  $\mathcal{A}(\mathcal{R})$  lies between the  $(1-\delta)k$ -level to the  $(1+\delta)k$ -level of  $\mathcal{A}(\mathcal{G})$ . This holds with high probability.

Namely, we have  $\mathbf{L}_{\mathcal{G},(1-\delta)k} \preceq \mathbf{L}_{\mathcal{R},\tilde{\kappa}} \preceq \mathbf{L}_{\mathcal{G},(1+\delta)k}$  and  $\mathbf{U}_{\mathcal{G},(1+\delta)k} \preceq \mathbf{U}_{\mathcal{R},\tilde{\kappa}} \preceq \mathbf{U}_{\mathcal{G},(1-\delta)k}$ .

*Proof:* This follows readily from the Chernoff inequality, due to space limitations we omit the proof. The interested reader can find it in the online full-version of this paper [Har05].

Lemma 4.2 suggests that instead of picking a specific level in a chunk of levels, as done in Section 3, we can instead pick a level, which is a shallow level of the appropriate random sample, and with high probability this level lies inside the allowable range. The only problem is that even this shallow level might (and will) have unreasonable complexity. We rectify this by doing direct approximation of the shallow levels.

**Definition 4.3** Let  $\mathcal{G}$  be a set of surfaces in  $\mathbb{R}^d$ . The  $(k, r)$ -*extent*  $\mathcal{G}|_r^k : \mathbb{R}^{d-1} \rightarrow \mathbb{R}$  is defined as the vertical distance between the  $r$ -level and the top  $k$ -level of  $\mathcal{A}(\mathcal{G})$ , i.e., for any  $x \in \mathbb{R}^{d-1}$ , we have

$$\mathcal{G}|_r^k(x) = \mathbf{U}_{\mathcal{G},k}(x) - \mathbf{L}_{\mathcal{G},r}(x).$$

**Definition 4.4** ([HW04]) Let  $\mathcal{F}$  be a set of non-negative functions defined over  $\mathbb{R}^{d-1}$ . A subset  $\mathcal{F}' \subseteq \mathcal{F}$  is  $(k, \varepsilon)$ -*sensitive* if for any  $r \leq k$  and  $x \in \mathbb{R}^{d-1}$ , we have

$$\mathbf{L}_{\mathcal{F},r}(x) \leq \mathbf{L}_{\mathcal{F}',r}(x) \leq \mathbf{L}_{\mathcal{F},r}(x) + \frac{\varepsilon}{2} \mathcal{F}|_r^k(x); \quad \text{and}$$

$$\mathbf{U}_{\mathcal{F},r}(x) - \frac{\varepsilon}{2} \mathcal{F}|_k^r(x) \leq \mathbf{U}_{\mathcal{F}',r}(x) \leq \mathbf{U}_{\mathcal{F},r}(x).$$

We need the following result of Har-Peled and Wang [HW04]. It states that for well behaved set of functions, one can find a small subset of the functions such that the vertical extent of the subset approximates the extents of the whole set. This holds only for “shallow” levels  $\leq k$ . In our application  $k$  is going to be about  $O(\varepsilon^{-2} \log n)$ . Here is the legalese:

**Theorem 4.5** ([HW04]) Let  $\mathcal{F} = \{p_1^{1/2}, \dots, p_n^{1/2}\}$  be a family of  $d$ -variate functions defined over  $\mathbb{R}^d$ , where  $p_i$  is a  $d$ -variate polynomial, for  $i = 1, \dots, n$ . Given  $k$  and  $0 < \varepsilon < 1$ , one can compute, in  $O(n + k/\varepsilon^{2s})$  time, a subset  $\mathcal{F}' \subseteq \mathcal{F}$ , such that, with high probability,  $\mathcal{F}'$  is  $(k, \varepsilon)$ -sensitive for  $\mathcal{F}$ , and  $|\mathcal{F}'| = O(k/\varepsilon^{2s})$ , where  $s$  is the linearization dimension of the polynomials of  $\mathcal{F}$ .

Intuitively, Theorem 4.5 states that shallow levels of depth at most  $k$ , has approximation of size polynomial in  $k$  and  $1/\varepsilon$ , and matching bottom/top  $k$  levels have their mutual distances preserved up to a small multiplicative factor.

**The construction.** We partition the levels of  $\mathcal{A}(\mathcal{G})$  into chunks, according to the algorithm of Section 3, setting  $m = O((\log n)/\varepsilon^2)$ . The first top/bottom  $m$  levels of  $\mathcal{A}(\mathcal{G})$  we approximate directly by computing a set  $\mathcal{S}_0$  which is  $(m, \varepsilon/20)$ -sensitive for  $\mathcal{G}$ , using Theorem 4.5. Next, compute the  $i$ th bottom (resp., top) level of  $\mathcal{A}(\mathcal{S}_0)$ , for  $i = 0, \dots, m$ , and let  $\gamma_i$  (resp,  $\eta_i$ ) denote those levels. We assign weight one for each such surface.

For every pair of chunks of levels  $L_i$  and  $R_i$  from Section 3, for  $i = m + 1, \dots, M$ , we compute an appropriate random sample  $\mathcal{R}_i$ . We remind the reader that  $L_i$  spans the range of levels from  $\alpha_{i-1} + 1$  to  $(1 + \varepsilon/10)\alpha_{i-1}$ , see Section 3. As such, if want to find a random level that falls inside this range, we need to set  $\delta = \varepsilon/40$  and  $k = (1 + \varepsilon/20)\alpha_{i-1}$ , and now apply Lemma 4.2, which results in a random set  $\mathcal{R}_i$ , such that level  $l_i = O(\varepsilon^{-2} \log n)$  of  $\mathcal{A}(\mathcal{R}_i)$  lies between level  $\alpha_{i-1} + 1$  and  $(1 + \varepsilon/10)\alpha_{i-1}$  of  $\mathcal{A}(\mathcal{G})$ . We now approximate the top  $l$  level and bottom  $l_i$  level of  $\mathcal{A}(\mathcal{R}_i)$  by applying Theorem 4.5 to  $\mathcal{R}_i$ . This results in a set  $\mathcal{S}_i$  of size  $O(\text{poly}(\log n, 1/\varepsilon))$  of surfaces, such that the extent of the top/bottom  $l_i$  levels of  $\mathcal{A}(\mathcal{S}_{i-M})$ , is an  $(\varepsilon/40)$ -approximation to the extent of the top/bottom  $l_i$  levels in  $\mathcal{A}(\mathcal{R}_i)$ . We extract the bottom  $l_i$  level and top  $l_i$  level of  $\mathcal{A}(\mathcal{R}_i)$ . Let the two resulting surfaces be denoted by  $\gamma_i$  and  $\eta_i$ , respectively, and assign them weight  $|R_i|$ , for  $i = m + 1, \dots, M$ .

Note, that  $\gamma_i$  and  $\eta_i$  no longer have constant complexity, but their complexity is bounded by  $O(\text{poly}(\log n, 1/\varepsilon))$ . Let  $\mathcal{H} = \{\gamma_1, \eta_1, \dots, \gamma_M, \eta_M\}$  be the resulting set of weighted surfaces, and observe that the complexity of the arrangement  $\mathcal{A}(\mathcal{H})$  is  $O(\text{poly}(\log n, 1/\varepsilon))$ . Furthermore, the analysis of Section 3, implies that  $\nu_{\mathcal{G}}(p) \approx_{\varepsilon} \nu_{\mathcal{H}}(p)$ , for any point  $p \in \mathbb{R}^d$ .

**Implementation details.** To get a linear running time, we need to carefully implement the above algorithm. First, observe that we computed  $O(\varepsilon^{-1} \log n)$  random samples  $\mathcal{R}_{m+1}, \dots, \mathcal{R}_M$ . Observe that if two random samples are generated by sampling every surface with probabilities which are similar (up to a factor of two), then we can just use the same random sample. Thus, we need to generate random samples only for probabilities which are powers of two (implying that only  $O(\log n)$  random samples are needed). In particular, let  $R_i$  be a random sample generated by by picking each surface of  $\mathcal{G}$  with probability  $1/2^i$ .

To perform this sampling quickly we generate the  $(i + 1)$ th random sample by picking each surface of  $R_i$  into  $R_{i+1}$  with probability half (the sequence of random samples  $\mathcal{G} = R_0, R_1, \dots, R_{O(\log n)}$  is sometimes referred to as a *gradation*). Namely, each  $R_i$  serves as a replacement for a sequence of random samples  $\mathcal{R}_{j_i}, \dots, \mathcal{R}_{j_{i+1}-1}$  which were generated using similar probabilities.

Next, we need to approximate the “shallow” levels of  $R_i$  up to depth  $O(\max(l_{j_i}, \dots, l_{j_{i+1}-1})) = O(\varepsilon^{-2} \log n)$ . Again, we are performing the computation of the shallow levels for a batch of samples using instead this single sample. Using Theorem 4.5, this takes  $O(|R_i| + \text{poly}(\log n, 1/\varepsilon))$  time. By the Chernoff inequality, with high probability, we have  $|R_i| = O(n/2^i)$ . Thus the overall running time, with high probability, is  $\sum_i O(n/2^i + \text{poly}(\log n, 1/\varepsilon)) = O(n + \text{poly}(1/\varepsilon, \log n))$ . Putting everything together, we have:

**Theorem 4.6** *Given a set of  $n$  unweighted surfaces  $\mathcal{G}$  in  $\mathbb{R}^d$ , as in Section 2.1.2, and a*

parameter  $\varepsilon$ , one can compute a set  $\mathcal{H}$  of surface patches, such that each patch is a portion of a surface of  $\mathcal{G}$  which is defined over (say) a simplex in  $\mathbb{R}^{d-1}$ . The number of surface patches is  $O(\text{poly}(1/\varepsilon, \log n))$ . Furthermore,  $\nu_{\mathcal{G}}(p) \approx \nu_{\mathcal{H}}(p)$  and  $\mu_{\mathcal{G}}(p) \approx \mu_{\mathcal{H}}(p)$ , for any point  $p \in \mathbb{R}^d$ . The algorithm takes  $O(n + \text{poly}(\log n, 1/\varepsilon))$  time and it succeeds with high probability.

Note, that total weight of the surface patches intersecting any vertical line  $\ell$  is equal to  $|\mathcal{G}|$ .

Theorem 4.6 shows that given an instance of any of the problems defined in Section 2.1.2, we can quickly reduce the problem size to a small weighted set of surface patches. This, while beneficial, still leaves us with the mundane task of solving the problem on the reduced instance. Since we no longer have to care too much about efficiency the problem becomes more manageable and we tackle it in the next section.

## 5 A Slow Approximation Algorithm

Due to space limitations, we only include here the main result of this section. For more details, see the online full-version of this paper [Har05]

**Theorem 5.1** *Let  $\mathcal{G}$  be a set of  $n$  weighted surface patches in  $\mathbb{R}^d$ , with linearization dimension  $s$ , such that any vertical line intersects surfaces with total weight  $W$ . Let  $\varepsilon > 0$  be a parameter. Then one can compute, in  $O(n^{3s+1}\varepsilon^{-s}\log^s W)$  time, a point  $x \in \mathbb{R}^d$ , such that  $\nu_{\mathcal{G}}(x) \leq (1 + \varepsilon)\nu_{\text{opt}}(\mathcal{G})$ , where  $\nu_{\text{opt}}(\mathcal{G}) = \min_{x \in \mathbb{R}^d} \nu_{\mathcal{G}}(x)$ .*

*One can also compute, in the same time complexity, a point  $y \in \mathbb{R}^d$ , such that  $\mu_{\mathcal{G}}(x) \leq (1 + \varepsilon)\mu_{\text{opt}}(\mathcal{G})$ , where  $\mu_{\text{opt}}(\mathcal{G}) = \min_{x \in \mathbb{R}^d} \mu_{\mathcal{G}}(x)$ .*

*Proof:* The correctness follows from the above discussion. As for the running time, there are  $O(n^{2s})$  prisms. In each prism we have  $n$  surfaces, and every surface get replicated  $O(\varepsilon^{-1}\log W)$  times. As such, the complexity of the arrangement inside each prism is  $O((n\varepsilon^{-1}\log W)^{s+1})$ . A careful implementation would require time proportional to the complexity of all those arrangements, which is  $O(n^{3s+1}\varepsilon^{-s}\log^s W)$ , as claimed. ■

## 6 The Main Result and Some Applications

By plugging Theorem 4.6 into Theorem 5.1, we get the main result of this paper:

**Theorem 6.1** *Given a set of  $n$  unweighted surfaces  $\mathcal{G}$  in  $\mathbb{R}^d$ , as defined in Section 2.1.2, and a parameter  $0 < \varepsilon < 1/4$ , then one can compute, in  $O(n + \text{poly}(\log n, 1/\varepsilon))$  time, a point  $x \in \mathbb{R}^d$ , such that  $\nu_{\mathcal{G}}(x) \leq (1 + \varepsilon)\nu_{\text{opt}}(\mathcal{G})$ , where  $\nu_{\text{opt}}(\mathcal{G}) = \min_{x \in \mathbb{R}^d} \nu_{\mathcal{G}}(x)$ .*

*One can also compute, in the same time complexity, a point  $y \in \mathbb{R}^d$ , such that  $\mu_{\mathcal{G}}(y) \leq (1 + \varepsilon)\mu_{\text{opt}}(\mathcal{G})$ , where  $\mu_{\text{opt}}(\mathcal{G}) = \min_{x \in \mathbb{R}^d} \mu_{\mathcal{G}}(x)$ .*

*The algorithm is randomized and succeeds with high probability.*

## 6.1 Applications

The discussion Section 2.1.1 implies that we can readily apply Theorem 6.1 to the problem of  $L_1$ -fitting a circle to a set of points in the plane. Note, that in fact the same reduction would work for the problem of  $\mu(\cdot)$  fitting, and for fitting a sphere to points in higher dimensions. We conclude:

**Theorem 6.2 ( $L_1/L_2$ -fitting points to a circle/sphere)** *Let  $P$  be a set of  $n$  points in  $\mathbb{R}^d$ , and  $\varepsilon > 0$  a parameter. One can  $(1 + \varepsilon)$ -approximate the sphere best fitting the points of  $P$ , where the price is the sum of Euclidean distances of the points of  $P$  to the sphere. The running time of the algorithm is  $O(n + \text{poly}(\log n, 1/\varepsilon))$ , and the algorithm succeeds with high probability.*

*Similarly, one can  $(1 + \varepsilon)$ -approximate the sphere that minimizes the sum of square distances of the points to the sphere.*

To our knowledge, Theorem 6.2 is the first subquadratic algorithm for this problem. A roughly quadratic time algorithm for the problem of  $L_1$ -fitting a circle to points in the plane was provided by Har-Peled and Koltun [HK04a]

### 6.1.1 $L_1/L_2$ -Fitting a cylinder to a point-set

Let  $P = \{p_1, \dots, p_n\}$  be a set of  $n$  points in  $\mathbb{R}^d$ ,  $\ell$  be a line in  $\mathbb{R}^d$  parameterized by a point  $q \in \ell$ , and a direction  $\vec{v}$  on the unit sphere  $\mathbb{S}^{(n)} \subseteq \mathbb{R}^d$ , and let  $r$  be the radius of the cylinder having  $\ell$  as its center. We denote by  $\mathfrak{C} = \mathfrak{C}(q, \vec{v}, r)$  the cylinder having  $\ell = \cup_{t \in \mathbb{R}} (q + t\vec{v})$  as its center. For a point  $p_i \in P$ , we have that its distance from  $\mathfrak{C}$  is

$$f_i(q, \vec{v}, r) = d(p_i, \mathfrak{C}) = \left| \|p_i - q - \langle p_i - q, \vec{v} \rangle \vec{v}\| - r \right| = \left| \sqrt{p_i(q, \vec{v}, r)} - r \right|,$$

where  $p_i(q, \vec{v}, r)$  is a polynomial with linearization dimension  $O(d^4)$  (as can be easily verified), for  $i = 1, \dots, n$ . The linearization dimension in this case can be reduced with more care, see [AHV04]. Thus, the overall price of fitting  $\mathfrak{C}$  to the points of  $P$  is  $\sum_i f_i(\mathfrak{C})$ . This falls into our framework, and we get:

**Theorem 6.3 ( $L_1/L_2$ -fitting points to a cylinder)** *Let  $P$  be a set of  $n$  points in  $\mathbb{R}^d$ , and  $\varepsilon > 0$  a parameter. One can  $(1 + \varepsilon)$ -approximate the cylinder that best fits the points of  $P$ , where the price is the sum of Euclidean distances of the points of  $P$  to the cylinder. The running time of the algorithm is  $O(n + \text{poly}(\log n, 1/\varepsilon))$ , and the algorithm succeeds with high probability.*

*Similarly, one can  $(1 + \varepsilon)$ -approximate the cylinder that minimizes the sum of square distances of the points of  $P$  to the cylinder.*

Interestingly, in two dimensions, the algorithm of Theorem 6.3 solves the problem of finding two parallel lines that minimizes the sum of distances of the points to the lines (i.e., each point contributes its distance to the closer of the two lines).

## 7 Conclusions

We had described in this paper a general approximation technique for problems of  $L_1$ -fitting of a shape to a set of points in low dimension. The running time of the new algorithm is  $O(n + \text{poly}(\log n, 1/\varepsilon))$ , which is a linear running time for a fixed  $\varepsilon$ . The constant powers hiding in the polylogarithmic term are too embarrassing to be explicitly stated, but are probably somewhere between 20 to 60 even just for the problem of  $L_1$ -fitting a circle to a set of points in the plane. Namely, this algorithm is only of theoretical interest. As such, the first open problem raised by this work is to improve these constants. A considerably more interesting problem is to develop a practical algorithm for this family of problems.

A natural tempting question is whether one can use the techniques in this paper, for the problem of  $L_1$ -fitting a spline or a Bezier curve to a set of points. Unfortunately, the resulting surfaces in the parametric space are no longer nice functions. As such, the algorithmic difficulty here is overshadowed by algebraic considerations. We leave this as an open problem for further research.

Another natural question is whether one can use the techniques of Har-Peled and Wang [HW04] directly, to compute a coresets for this problem, and solve the problem on the coresets directly (our solution did a similar thing, by breaking the parametric space into a small number of prisms, and constructing a small “sketch” inside each such region). There is unfortunately a nasty technicality that requires that a coresets for the  $L_1$ -fitting of linear function, is also coresets if we take the square root of the functions (as holds for the construction of Section 3). It seems doubtful that this claim holds in general, but maybe a more careful construction of a coresets for the planes case would still work. The author leaves this as an open problem for further research.

The author believes that the algorithm presented in this paper should have a lot of other applications. We leave this as an open problem of further research.

## Acknowledgments

The author thanks Pankaj Agarwal, Arash Farzan, Vladlen Koltun, Bardia Sadri, Kasturi Varadarajan, and Yusu Wang for useful and insightful discussions related to the problems studied in this paper.

## References

- [AAHS00] P. K. Agarwal, B. Aronov, S. Har-Peled, and M. Sharir. Approximation and exact algorithms for minimum-width annuli and shells. *Discrete Comput. Geom.*, 24(4):687–705, 2000.
- [AHV04] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Approximating extent measures of points. *J. Assoc. Comput. Mach.*, 51(4):606–635, 2004.
- [AHY06] P. Agarwal, S. Har-Peled, and H. Yu. Robust shape fitting via peeling and grating coresets. In *Proc. 17th ACM-SIAM Sympos. Discrete Algorithms*, 2006. to appear.

- [AM94] P. K. Agarwal and J. Matoušek. On range searching with semialgebraic sets. *Discrete Comput. Geom.*, 11:393–418, 1994.
- [BH01] G. Barequet and S. Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. *J. Algorithms*, 38:91–109, 2001.
- [Cha02] T. M. Chan. Approximating the diameter, width, smallest enclosing cylinder and minimum-width annulus. *Internat. J. Comput. Geom. Appl.*, 12(2):67–85, 2002.
- [Cla05] K. L. Clarkson. Subgradient and sampling algorithms for l1 regression. In *Proc. 16th ACM-SIAM Sympos. Discrete Algorithms*, 2005. to appear.
- [FKV98] A. Frieze, R. Kannan, and S. Vempala. Fast monte-carlo algorithms for finding low-rank approximations. In *FOCS '98: Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, page 370. IEEE Computer Society, 1998.
- [Har05] S. Har-Peled. How to get close to the median shape. Available from [http://www.uiuc.edu/~sariel/papers/05/l1\\_fitting/](http://www.uiuc.edu/~sariel/papers/05/l1_fitting/), 2005.
- [HK04a] S. Har-Peled and V. Koltun. Approximate  $l_1$  and  $l_2$  circle fitting in (easy) polynomial time. manuscript, 2004.
- [HK04b] S. Har-Peled and A. Kushal. Smaller coresets for  $k$ -median and  $k$ -means clustering. [http://www.uiuc.edu/~sariel/papers/04/small\\_coreset/](http://www.uiuc.edu/~sariel/papers/04/small_coreset/), 2004.
- [HM04] S. Har-Peled and S. Mazumdar. Coresets for  $k$ -means and  $k$ -median clustering and their applications. In *Proc. 36th Annu. ACM Sympos. Theory Comput.*, pages 291–300, 2004.
- [HW04] S. Har-Peled and Y. Wang. Shape fitting with outliers. *SIAM J. Comput.*, 33(2):269–285, 2004.
- [RVW04] L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via adaptive sampling. manuscript, 2004.
- [SW89] G.A.F. Seber and C.J. Wild. *Nonlinear regression*. Jonh Wiley & Sons, 1989.
- [YKII88] P. Yamamoto, K. Kato, K. Imai, and H. Imai. Algorithms for vertical and orthogonal l1 linear approximation of points. In *Proc. 4th Annu. ACM Sympos. Comput. Geom.*, pages 352–361. ACM Press, 1988.
- [ZS02] Y. Zhou and S. Suri. Algorithms for a minimum volume enclosing simplex in three dimensions. *SIAM J. Comput.*, 31(5):1339–1357, 2002.