

# Efficiently Approximating the Minimum-Volume Bounding Box of a Point Set in Three Dimensions \*

Gill Barequet  
Faculty of Computer Science,  
The Technion—IIT,  
Haifa 32000, Israel

Sariel Har-Peled  
Department of Computer  
Science, DCL 2111; University of Illinois;  
1304 West Springfield Ave.; Urbana, IL 61801; USA

June 30, 2001

## Abstract

We present an efficient  $O(n + 1/\varepsilon^{4.5})$ -time algorithm for computing a  $(1 + \varepsilon)$ -approximation of the minimum-volume bounding box of  $n$  points in  $\mathbb{R}^3$ . We also present a simpler algorithm (for the same purpose) whose running time is  $O(n \log n + n/\varepsilon^3)$ . We give some experimental results with implementations of various variants of the second algorithm. The implementation of the algorithm described in this paper is available online [Har00].

## 1 Introduction

In this paper we give efficient algorithms for solving the following problem:

Given a set  $S$  of  $n$  points in  $\mathbb{R}^3$  and a parameter  $0 < \varepsilon \leq 1$ , find a box that encloses  $S$  and approximates the minimum-volume bounding box of  $S$  by a factor  $(1 + \varepsilon)$ .

---

\*Work on this paper by the first author was done while he was affiliated with the Center for Geometric Computing, Dept. of Computer Science, Johns Hopkins University, Baltimore, MD 21218. A preliminary version of this paper appeared in [BH99], and the final version appeared in [BH01]. Work of the first author has been supported by the U.S. Army Research Office under Grant DAAH04-96-1-0013. Work of the second author has been supported by a grant from the U.S.-Israeli Binational Science Foundation.

We are not aware to any previously-published algorithm that solves this problem.

Three-dimensional boxes which enclose sets of points are used for maintaining hierarchical partitioning of sets of points. These data structures have important applications in computer graphics (e.g., for fast rendering of a scene or for collision detection), statistics (for storing and performing range-search queries on a large database of samples), etc. From a top-down viewpoint, the problem in such applications is divided into two (admittedly, related) problems of splitting a given set of points into two (or more) subsets, and of computing a nearly-optimal box (or another generic shape) that encloses each subset. In this paper we concentrate on the second problem.

Numerous heuristics have been proposed for computing a box which encloses a given set of points. The simplest heuristic was naturally to compute the axis-aligned bounding box (AABB) of the point set. Two-dimensional variants of this heuristic include the well-known R-tree, the *packed* R-tree [RL85], the R<sup>+</sup>-tree [SRF87], the R\*-tree [BKSS90], etc. [HKM95] use a minimum-volume AABB trimmed in a fixed number of directions for speeding up collision detection. [GLM96] implement in their RAPID system the OBB-tree (a tree of arbitrarily-oriented bounding boxes), where each box which encloses a set of polygons is aligned with the *principal components* of the distribution of polygon vertices. A similar idea is used by [BCG<sup>+</sup>96] for the BOXTREE. The latter work suggests a few variants, in which the computed box is aligned with only one principal component of the point distribution (e.g., the one that corresponds to the smallest or largest eigenvalue of the covariance matrix of the point coordinates), and the other two directions are determined by another method (e.g., by computing the exact minimum-area bounding rectangle of the projection of the points into a plane orthogonal to the first chosen direction). Other generic shapes, such as a sphere [Hub95], a cone [Sam89], or a prism [FP87, BCG<sup>+</sup>96] were also used for maintaining a hierarchical data structure of point containers. Most of these heuristics require  $O(n)$  time and space for computing the bounding box (or another shape) but do not provide a guaranteed value (approximation factor of the optimum) of the output.

An algorithm of [BS97] solves a similar problem, in which the  $n$  points are to be contained in two axis-aligned boxes, and the goal is to minimize the volume (or any other monotone measure) of the larger box. Their algorithm requires  $O(n^2)$  time.

O'Rourke presented the only algorithm (to the best of our knowledge) for computing the exact arbitrarily-oriented minimum-volume bounding box of a set of  $n$  points in  $\mathbb{R}^3$ . His algorithm requires  $O(n^3)$  time and  $O(n)$  space. In this paper we present the first two  $(1 + \varepsilon)$ -approximation algorithms for the minimum-volume bounding-box problem. Both algorithms are nearly linear in  $n$ . The running times of these algorithms are  $O(n + 1/\varepsilon^{4.5})$  and  $O(n \log n + n/\varepsilon^3)$ .

The paper is organized as follows. In Section 2 we give the notations and definitions used throughout the paper. In Sections 3 and 4 we present the two  $(1 + \varepsilon)$ -approximation bounding-box approximations algorithms. In Section 5 we presents experimental results. We end in Section 6 with some concluding remarks.

## 2 Notations and Definitions

We first present the notation used in this paper.

We denote the origin of coordinates by  $o$  and the unit cube by  $\mathcal{C}$ . Throughout the paper we denote by  $Q$  a point set in two dimensions, and use  $S$  to denote a similar set in three dimensions. Unless specified otherwise, the set is assumed to contain  $n$  (or a comparable number of) points. We denote by  $\mathcal{CH}(Q)$  (resp.,  $\mathcal{CH}(S)$ ) the 2-dimensional (resp., 3-dimensional) convex hull of  $Q$  (resp.,  $S$ ). The symbol  $P$  is used for denoting a convex polyhedron in three-space.

The symbols  $R$  and  $B$  are used for rectangles and boxes, respectively. The notation  $B(S)$  is used for any bounding box of a point set  $S$ . We also use the notation  $B = (b_1, b_2, b_3)$ , where  $b_1, b_2, b_3$  are three orthogonal vectors in  $\mathbb{R}^3$  having the directions and sizes of the edges of  $B$ . The operators  $\text{Area}(R)$  and  $\text{Vol}(B)$  denote the area of  $R$  and the volume of  $B$ , respectively.

We denote by  $R_{\text{opt}}(Q)$  the *minimum-area bounding rectangle* of  $Q$  and by  $B_{\text{opt}}(S)$  the *minimum-volume bounding box* of  $S$ .<sup>1</sup> Let  $V$  be a set of orthogonal vectors in  $\mathbb{R}^3$ . We denote by  $B_{\text{opt}}(S, V)$  for the minimum-volume bounding box of  $S$  whose set of directions contains  $V$ . We denote some constant approximation of the minimum-volume bounding box of  $S$  (for some predefined positive constant) by  $B^*(S)$ . (Such a box is computed in Lemmas 3.4 and 3.6—see Section 3.)

Finally, given a box  $B = (b_1, b_2, b_3)$  in  $\mathbb{R}^3$ , the grid of points spanned by  $B$  is

$$\text{Grid}(B) = \left\{ i_1 b_1 + i_2 b_2 + i_3 b_3 \mid i_1, i_2, i_3 \in \mathbb{Z} \right\}.$$

Let  $\mathcal{G} = \text{Grid}(B)$ . Denote by

$$B_{(i,j,k)}^{\mathcal{G}} = \left\{ x_1 b_1 + x_2 b_2 + x_3 b_3 \mid \begin{array}{l} i \leq x_1 \leq i+1, \\ j \leq x_2 \leq j+1, \\ k \leq x_3 \leq k+1, \end{array} i, j, k \in \mathbb{Z} \right\}$$

the  $(i, j, k)$ -th cell of  $\mathcal{G}$ . For a prescribed constant  $d > 0$ , let

$$G(B, d) = \left\{ i_1 b_1 + i_2 b_2 + i_3 b_3 \mid i_1, i_2, i_3 \in \mathbb{Z}, |i_1|, |i_2|, |i_3| \leq d \right\}$$

be the set of points of  $\mathcal{G}$  whose  $L_\infty$ -distance (along the directions  $b_1, b_2, b_3$ ) from  $o$  is at most  $d$ .

### 3 An Efficient Approximation Algorithm

In this section we present our main approximation algorithm to the minimum-volume bounding box of a set of points in three dimensions.

#### 3.1 Approximating the Diameter

First we need a good-enough approximation of the diameter of the point set.

---

<sup>1</sup>Since we are interested in the *area* (or *volume*) of the object, we may pick an arbitrary minimum rectangle (or box) if several exist.

**Definition 3.1** The *diameter* of a point set  $S \in \mathbb{R}^3$  (denoted by  $\mathcal{D}(S)$ ) is the distance between the two furthest points of  $S$ . That is,  $\mathcal{D}(S) = \max_{s,t \in S} |st|$ .

We can easily find a pair of points in  $S$  whose mutual distance is a  $1/\sqrt{d}$ -approximation of the diameter of  $S$ :<sup>2</sup>

**Lemma 3.2** *Given a point set  $S$  in  $\mathbb{R}^d$  (for a fixed  $d$ ), one can compute in  $O(n)$  time a pair of points  $s, t \in S$ , such that  $|st| \leq \mathcal{D}(S) \leq \sqrt{d}|st|$ .*

*Proof:* Let  $B$  be the minimum axis-parallel box containing  $S$ , and let  $s$  and  $t$  be the points in  $S$  that define the longest edge of  $B$ , whose length is denoted by  $l$ . By the diameter defn,  $|st| \leq \mathcal{D}(S)$ , and clearly,  $\mathcal{D}(S) \leq \sqrt{d}l \leq \sqrt{d}|st|$ . The points  $s$  and  $t$  are easily found in  $O(nd)$  time. Since  $d$  is fixed, this is actually  $O(n)$  time. ■

In particular, we can approximate in linear time the diameter of a point set in three dimensions by a factor of  $1/\sqrt{3}$ . (In fact, one can find in linear time a  $(1/\sqrt{3})$ -approximation of the diameter in *any* dimension; see [EK89].)

Actually, we can find any arbitrarily-good approximation of the diameter. Since we were not able to find any reference with a proof of the following folklore lemma (see [Har99] for a similar result), we include an easy proof of it here.

**Lemma 3.3** *Given a set  $S$  of  $n$  points in  $\mathbb{R}^d$  (for a fixed  $d$ ) and  $\varepsilon > 0$ , one can compute in  $O(n + 1/\varepsilon^{2(d-1)})$  time a pair of points  $s, t \in S$  such that  $|st| \geq (1 - \varepsilon)\mathcal{D}(S)$ .*

*Proof:* Let  $B$  be the minimum axis-parallel box containing  $S$ , and let  $\mathcal{G} = \text{Grid}((\varepsilon/(2\sqrt{d}))B)$ . For a point  $x \in \mathbb{R}^d$ , denote by  $x_{\mathcal{G}}$  the closest point of  $\mathcal{G}$  to  $x$ . Define  $S_{\mathcal{G}} = \{x_{\mathcal{G}} \mid x \in S\}$ . Finally, let  $l$  be the length of the longest diagonal of a cell of  $\mathcal{G}$ . Clearly,  $l \leq (\varepsilon/2)\mathcal{D}(S)$ . For every pair of points  $x, y \in S$  we have

$$|xy| - l \leq |x_{\mathcal{G}}y_{\mathcal{G}}| \leq |xy| + l.$$

Thus,  $\mathcal{D}(S_{\mathcal{G}}) \geq \mathcal{D}(S) - l$ . Let  $s, t$  be the two points in  $S$  for which  $s_{\mathcal{G}}, t_{\mathcal{G}}$  realize the diameter of  $S_{\mathcal{G}}$ . Now,

$$|st| \geq |s_{\mathcal{G}}t_{\mathcal{G}}| - l = \mathcal{D}(S_{\mathcal{G}}) - l \geq \mathcal{D}(S) - 2l \geq (1 - \varepsilon)\mathcal{D}(S).$$

The set  $S_{\mathcal{G}}$  can be computed in  $O(n)$  time (where the hidden constant of proportionality contains a factor  $d$ ). The cardinality of  $S_{\mathcal{G}}$  is  $O(1/\varepsilon^d)$ . Thus, we can compute the diameter of  $S_{\mathcal{G}}$  (in a brute force manner) in  $O(1/\varepsilon^{2d})$  time. The two points of  $S$  that correspond to the diameter of  $S_{\mathcal{G}}$  are, by the above analysis, the sought  $(1 - \varepsilon)$ -approximation of the diameter of  $S$ .

Any point of  $S_{\mathcal{G}}$  that lies between two other points of  $S_{\mathcal{G}}$  (along one of the axes) can not correspond to a point of  $S$  that defines  $\mathcal{D}(S)$ . By removing all such points in  $O(n)$  time we can consider only  $O(1/\varepsilon^{d-1})$  points. Hence the running time of the algorithm is improved to  $O(n + 1/\varepsilon^{2(d-1)})$ . ■

The two points of  $S_{\mathcal{G}}$  that realize its diameter must be vertices of the convex-hull of  $S_{\mathcal{G}}$ . Thus, the running time of the algorithm of Lemma 3.3 can be further improved by first

---

<sup>2</sup> Note that we approximate the diameter from *below*, while we approximate the minimum-volume bounding box from *above*.

computing the set of vertices of  $\mathcal{CH}(S_G)$ , denoted by  $S_{\mathcal{CH}(S_G)}$ . Set  $h = |S_{\mathcal{CH}(S_G)}|$ . [And63] showed that  $h = O(1/\varepsilon^{(d-1)d/(d+1)})$ . For  $d \leq 3$  we compute  $S_{\mathcal{CH}(S_G)}$  by computing the entire convex-hull of  $S_G$  in  $O(|S_G| \log |S_G|)$  time. For higher dimensions we use an output-sensitive algorithm [Cha96]. Let  $m$  denote the number of points in  $S_G$ . Clearly,  $m = O(1/\varepsilon^{(d-1)})$ . The time required for computing  $S_G$  is

$$O\left(m \log^{d+2} h + (mh)^{1-\frac{1}{\lfloor d/2 \rfloor + 1}} \log^{O(1)} m\right) = O\left(m^{\frac{2d}{d+1}}\right) = O\left(\left(\frac{1}{\varepsilon}\right)^{\frac{2d(d-1)}{d+1}}\right).$$

Computing the diameter of  $S_{\mathcal{CH}(S_G)}$  (in a brute-force manner) requires  $O(h^2) = O\left((1/\varepsilon)^{\frac{2d(d-1)}{d+1}}\right)$  time. Overall, the running time of the algorithm of Lemma 3.3 can be improved to  $O\left((1/\varepsilon)^{\frac{2d(d-1)}{d+1}}\right)$ .

We can do even better in three dimensions if we are willing to sacrifice simplicity. In this case we compute the exact diameter of  $\mathcal{CH}(S_G)$  in  $O((1/\varepsilon^{3/2}) \log(1/\varepsilon))$  time [CS89]. Overall, we compute a  $(1 - \varepsilon)$ -approximation of the diameter of  $S \in \mathbb{R}^3$  in  $O(n + (1/\varepsilon^{3/2}) \log(1/\varepsilon))$  time.

## 3.2 Computing an Approximating Box

Let  $Q$  be a set of  $n$  points in  $\mathbb{R}^2$ . Computing  $R_{\text{opt}}(Q)$  can be done in  $O(n \log n)$  time [Tou83]. (Hence, given a set  $S$  of  $n$  points and a direction  $v$  in  $\mathbb{R}^3$ , one can compute  $B_{\text{opt}}(S, \{v\})$  in  $O(n \log n)$  time.) The bottleneck of the cited algorithm is the computation of  $\mathcal{CH}(Q)$ ; when the latter is given in advance,  $R_{\text{opt}}(Q)$  can be computed in optimal  $\Theta(n)$  time.

**Lemma 3.4** *Given a set  $S$  of  $n$  points in  $\mathbb{R}^3$ , one can compute in  $O(n)$  time a bounding box  $B(S)$  with  $\text{Vol}(B_{\text{opt}}(S)) \leq \text{Vol}(B(S)) \leq 6\sqrt{6} \text{Vol}(B_{\text{opt}}(S))$ .*

*Proof:* By using the algorithm of Lemma 3.2 we compute in  $O(n)$  time two points  $s, t \in S$  which form a  $(1/\sqrt{3})$ -approximation of the diameter of  $S$ . Let  $H$  be a plane perpendicular to  $st$ , and let  $Q$  be the orthogonal projection of  $S$  into  $H$ .

Now, by using the algorithm of Lemma 3.2 again, we compute in  $O(n)$  time two points  $s', t' \in Q$  (see Figure 1) for which  $\sqrt{2}|s't'| \geq \mathcal{D}(Q)$ . Let  $\mu$  be a direction perpendicular to  $st$  and  $s't'$ . We claim that the box  $B^* = B_{\text{opt}}(S, \{st, s't', \mu\})$  is a  $(6\sqrt{6})$ -approximation of  $B_{\text{opt}}(S)$ .

Indeed, let  $R$  be the minimum-area bounding rectangle of  $Q$  in the directions  $s't'$  and  $\mu$ , let  $\omega$  be the length of the edge of  $R$  in the direction  $\mu$ , and let  $u, v$  be the two points of  $Q$  lying on the two edges of  $R$  parallel to  $s't'$ . Clearly,

$$\text{Area}(R) \leq \omega \mathcal{D}(Q) \leq \sqrt{2} \omega |s't'|.$$

On the other hand, the quadrilateral  $F = s'ut'v$  is contained in  $\mathcal{CH}(Q)$ , and its area is  $|s't'| \omega / 2$ .

Set  $\mathcal{S} = \mathcal{CH}(S)$ . Let  $s_H$  be the orthogonal projection of  $s$  into  $H$ , and let  $T_1, \dots, T_m$  be a triangulation of  $\mathcal{CH}(Q)$  (within  $H$ ) in which all the triangles share the vertex  $s_H$ .

We next show that  $\text{Vol}(\mathcal{S}) \geq \text{Area}(\mathcal{CH}(Q)) |st| / 3$ . The outer edge  $p'_i q'_i$  of  $T_i$  is a projection of a pair of points  $p_i, q_i \in S$  (for  $i = 1, \dots, m$ ) into  $H$ . Thus, each triangle  $T_i$  corresponds

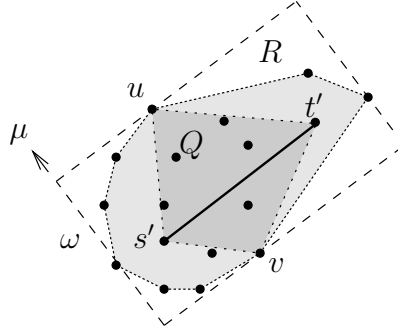


Figure 1: The convex hull of the projection of  $S$  contains a large quadrangle

to a tetrahedron  $\mathcal{T}_i = \mathcal{CH}(\{p_i, q_i, s, t\})$  (for  $i = 1, \dots, m$ ). These tetrahedra are pairwise disjoint in their interiors, and

$$\text{Vol}(\mathcal{T}_i) = \text{Vol}(\mathcal{CH}(\{p_i, q_i, s, t\})) = \frac{\text{Area}(\mathcal{T}_i)|st|}{3},$$

where we use the fact that the volume of a tetrahedron does not change when we translate one of its vertices  $u$  in a direction parallel to an edge (of the tetrahedron) that does not coincide with  $u$  (or, in more generality, parallel to the opposite face of  $u$ ). We thus have

$$\begin{aligned} \frac{|s't'|\omega}{6} &= \frac{\text{Area}(F)|st|}{3} \leq \frac{\text{Area}(\mathcal{CH}(Q))|st|}{3} \\ &= \frac{(\sum_{i=1}^m \text{Area}(\mathcal{T}_i))|st|}{3} = \sum_{i=1}^m \text{Vol}(\mathcal{T}_i) \leq \text{Vol}(\mathcal{S}). \end{aligned}$$

On the other hand,

$$\text{Vol}(\mathcal{S}) \leq \text{Vol}(B^*) \leq \text{Area}(R)\mathcal{D}(S) \leq \sqrt{2}\omega|s't'|\sqrt{3}|st|.$$

Since  $S \subseteq B^*$ , we have  $\text{Vol}(\mathcal{S}) \leq \text{Vol}(B^*) \leq 6\sqrt{6} \text{Vol}(\mathcal{S})$ , as required.  $\blacksquare$

Note that the approximation factor for the diameter obtained in Lemma 3.4 can be improved by using better approximation schemes, e.g., the algorithm of Lemma 3.3. In any case a better constant approximation factor will be reflected by a higher constant hidden in the big-Oh notation of the running time.

The algorithm of Lemma 3.4 can be extended to a  $d$ -dimensional space by choosing the direction of the exact diameter  $v$  of the point set  $S$  as one direction of the bounding box, projecting the points to a hyperplane  $H \in \mathbb{R}^{d-1}$  perpendicular to  $v$ , solving recursively the  $(d-1)$ -dimensional problem in  $H$ , and outputting the Cartesian product of the  $(d-1)$ -dimensional solution and  $v$ . The volume of the computed box is at most  $d!$  times the volume of the optimal (minimum-volume) bounding box of  $S$ .<sup>3</sup> Moreover, the side lengths of the bounding box are found in decreasing order. [FL95] use a similar method for visualizing a

<sup>3</sup> Here we can trade time for approximation quality. By investing less time we can compute a  $(1/\sqrt{2})$ - (resp.,  $(1/\sqrt{3})$ -) approximation of the diameter in two (resp.,  $k$ , for  $3 \leq k \leq d$ ) dimensions, and obtain a  $(\sqrt{2} \cdot 3^{(d-2)/2} d!)$ -approximation of  $B_{\text{opt}}(S)$ . By investing even less time we can compute a  $(1/\sqrt{k})$ -approximation of the diameter in  $k$  dimensions (for  $2 \leq k \leq d$ ), and obtain a  $(d!)^{3/2}$ -approximation of  $B_{\text{opt}}(S)$ .

set of points in a high-dimensional space. They compute in a similar manner the first two (or three) directions, project the points into the subspace spanned by these directions, and display the projected points in this subspace.

We next show that a large-enough convex polyhedron contained in the unit cube  $\mathcal{C}$  must contain a large-enough axis-parallel cube.

**Lemma 3.5** *Let  $P$  be a convex polyhedron with the properties that  $P \subseteq \mathcal{C}$  and  $\text{Vol}(P) \geq 1/15$ . Then there exists a translation  $v \in \mathbb{R}^3$  for which  $\frac{1}{107}\mathcal{C} + v \subseteq P$ .*

*Proof:* It is easy to verify that the area of the intersection of  $\mathcal{C}$  with any plane is at most  $3\pi/4$ . (Indeed, the intersection area is maximized when the plane passes through the center of  $\mathcal{C}$ , and such a cross-section is contained in a disk of radius  $\sqrt{3}/2$ .) Hence,

$$\text{Width}(P) \geq \text{Vol}(P) / \left( \frac{3\pi}{4} \right) \geq \frac{4}{45\pi},$$

where  $\text{Width}(P)$  is the minimum distance between two parallel planes supporting  $P$ . Let  $r(P)$  be the radius of the largest ball  $K$  inscribed in  $P$ . It is known [GK92] that  $r(P) \geq \text{Width}(P)/(2\sqrt{3})$ . This implies  $r(P) \geq 2/(45\sqrt{3}\pi)$ . Finally,  $K$  inscribes an axis-parallel cube  $\mathcal{C}'$  whose side is of length  $(2/\sqrt{3})r(P) \geq 1/107$ , and  $\mathcal{C}' \subseteq K \subseteq P$ , as asserted. ■

Let  $A$  and  $B$  be two sets in  $\mathbb{R}^3$ . The *Minkowski sum* of  $A$  and  $B$  is the set  $A \oplus B = \{a + b \mid a \in A, b \in B\}$ . Given a constant  $c > 0$  and a box  $B \in \mathbb{R}^3$ , it is easily verified that  $B \oplus (c \cdot B)$  is also a box whose volume is  $(1 + c)^3 \text{Vol}(B)$ .

The idea of our main algorithm is approximating  $\mathcal{CH}(S)$  by a low-complexity convex polyhedron  $P \supseteq \mathcal{CH}(S)$ , followed by computing (exactly)  $B_{\text{opt}}(P)$ . The polyhedron  $P$  is chosen such that  $B_{\text{opt}}(P)$  is an  $(1 + \varepsilon)$ -approximation of  $B_{\text{opt}}(S)$ .

First we need a combined version of Lemmas 3.4 and 3.5:

**Lemma 3.6** *Let  $S$  be a set of  $n$  points in  $\mathbb{R}^3$ . One can compute in  $O(n)$  time a bounding box  $B^*(S)$  such that  $\text{Vol}(B_{\text{opt}}(S)) \leq \text{Vol}(B^*(S)) \leq 15 \text{Vol}(B_{\text{opt}}(S))$  and there exists a translation  $v \in \mathbb{R}^3$  for which  $\frac{1}{107}B^*(S) + v \subseteq \mathcal{CH}(S)$ .*

*Proof:* Let  $B = B^*(S)$  be the bounding box of  $S$  computed by Lemma 3.4. Let  $T$  and  $t$  be a nonsingular linear transformation and a translation, respectively, such that  $T(B) + t = \mathcal{C}$ . Careful observation of the construction in Lemma 3.4 shows that  $\text{Vol}(T(\mathcal{CH}(S))) \geq \frac{1}{15}$ . Hence by Lemma 3.5 there exists a translation  $v'$  such that  $\frac{1}{107}(T(B) + t) + v' \subseteq T(\mathcal{CH}(S)) + t$ , where we use the fact that linear transformations preserve volume order. Therefore  $\frac{1}{107}B + T^{-1}(v' - \frac{106}{107}t) \subseteq \mathcal{CH}(S)$ , and the claim follows. ■

Note that this lemma is true even if  $S$  is a planar set, in which case the minimum-area bounding rectangle provided by Lemma 3.4 degenerates to a segment, and the volume of the computed bounding box is 0. This guarantees that the approximation algorithm (described below) produces a degenerate box (of volume 0) for a degenerate (planar) input point set.

We are now ready to present the approximation algorithm for  $B_{\text{opt}}(S)$ .

Let  $B = B^*(S)$  be the bounding box of  $S$  computed by Lemma 3.6, and let  $B_\varepsilon$  be a translated copy of  $\frac{\varepsilon}{428}B$  centered at  $o$ . In addition, define  $\mathcal{S}^\oplus = \mathcal{CH}(S) \oplus B_\varepsilon$  and  $\mathcal{G} = \text{Grid}(\frac{1}{2}B_\varepsilon)$ . We approximate  $S$  on  $\mathcal{G}$ . For each point  $p \in S$  let  $\mathcal{G}(p)$  be the set of eight

vertices of the cell of  $\mathcal{G}$  that contains  $p$ , and let  $S_{\mathcal{G}} = \cup_{p \in S} \mathcal{G}(p)$ . Define  $P = \mathcal{CH}(S_{\mathcal{G}})$ . Clearly,  $\mathcal{CH}(S) \subseteq P \subseteq \mathcal{S}^{\oplus}$ . Moreover, one can compute  $P$  in  $O(n + (1/\varepsilon^2) \log(1/\varepsilon))$  time. On the other hand,  $P \subseteq B \oplus B_{\varepsilon}$ . The latter term is a box which contains at most  $k = 860/\varepsilon + 4$  grid points along each of the directions set by  $B$ , so  $k$  is also an upper bound for the number of grid points contained by  $P$  in each direction. [And63] showed that the complexity of  $P$  is  $O(k^{3/2}) = O(1/\varepsilon^{3/2})$ . We exploit this result in the analysis of the running time of the algorithm. Finally, we compute  $B_{\text{opt}}(P)$  exactly.

It remains to show that  $B_{\text{opt}}(P)$  is a  $(1 + \varepsilon)$ -approximation of  $B_{\text{opt}}(S)$ . Let  $B_{\text{opt}}^{\varepsilon}$  be a translation of  $\frac{\varepsilon}{4} B_{\text{opt}}(S)$  that contains  $B_{\varepsilon}$ . (The existence of  $B_{\text{opt}}^{\varepsilon}$  is guaranteed by Lemma 3.6.) Thus,  $P \subseteq \mathcal{CH}(S) \oplus B_{\varepsilon} \subseteq \mathcal{CH}(S) \oplus B_{\text{opt}}^{\varepsilon} \subseteq B_{\text{opt}}(S) \oplus B_{\text{opt}}^{\varepsilon}$ . Since  $B_{\text{opt}}(S) \oplus B_{\text{opt}}^{\varepsilon}$  is a box, it is a bounding box of  $P$  and therefore also of  $\mathcal{CH}(S)$ . Its volume is

$$\text{Vol}(B_{\text{opt}}(S) \oplus B_{\text{opt}}^{\varepsilon}) = \left(1 + \frac{\varepsilon}{4}\right)^3 \text{Vol}(B_{\text{opt}}(S)) < (1 + \varepsilon) \text{Vol}(B_{\text{opt}}(S)),$$

as desired. (The last inequality is the only place where we use the assumption  $\varepsilon \leq 1$ .)

To recap, the algorithm consists of the four following steps:

1. Compute the box  $B^*(S)$  (see Lemma 3.6) in  $O(n)$  time.
2. Compute the point set  $S_{\mathcal{G}}$  in  $O(n)$  time.
3. Compute  $P = \mathcal{CH}(S_{\mathcal{G}})$  in  $O(n + (1/\varepsilon^2) \log(1/\varepsilon))$  time. This is done by computing the convex hull of all the extreme points of  $S_{\mathcal{G}}$  along vertical lines of  $\mathcal{G}$ . We have  $O(1/\varepsilon^2)$  such points, thus computing their convex hull takes  $O((1/\varepsilon^2) \log(1/\varepsilon))$  time.
4. Compute  $B_{\text{opt}}(P)$  by the algorithm of [O'R85]. The complexity of  $P$  is  $O(1/\varepsilon^{3/2})$  [And63], so this step requires  $O((1/\varepsilon^{3/2})^3) = O(1/\varepsilon^{4.5})$  time.

Thus we obtain our main result:

**Theorem 3.7** *Let  $S$  be a set of  $n$  points in  $\mathbb{R}^3$ , and let  $0 < \varepsilon \leq 1$  be a parameter. One can compute in  $O(n + 1/\varepsilon^{4.5})$  time a bounding box  $B(S)$  with  $\text{Vol}(B(S)) \leq (1 + \varepsilon) \text{Vol}(B_{\text{opt}}(S))$ .*

Note that the box  $B(S)$  computed by the above algorithm is most likely not minimal along its directions. The minimum bounding box of  $S$  homothet of  $B(S)$  can be computed in additional  $O(n)$  time.

## 4 An Alternative Practical Algorithm

Unfortunately, the algorithm described in the previous section is perhaps too difficult to implement. In this section we suggest an asymptotically less efficient, but easier to implement, approximation algorithm for the minimum-volume bounding box problem.

In the algorithm described above we chose an approximation of the diameter of the set  $S$  as a “favorable” direction  $v$  and computed  $B_{\text{opt}}(S, \{v\})$ , which served for the defn of the grid  $\mathcal{G}$ . Then we expanded  $\mathcal{CH}(S)$  (which was not computed explicitly) into a low-complexity



**ALGORITHM** GRIDSEARCHMINVOLBBX ( $S, \varepsilon$ )  
**Input:** A set  $S$  of  $n$  points in  $\mathbb{R}^3$ , and a parameter  $0 < \varepsilon \leq 1$ .  
**Output:** A  $(1 + \varepsilon)$ -approximation of  $B_{\text{opt}}(S)$ .  
**begin**  
  Compute  $\mathcal{CH}(S)$ ;  
  Compute  $B^*(S)$ ; /\* The box generated by Lemma 3.6 \*/  
  Compute  $\text{BG} = G(B^*(S), c/\varepsilon)$ ; /\* Refer to the text for the value of  $c$  \*/  
  Set  $\text{min\_vol} := \infty$  and  $v^* := \text{undefined}$ ;  
  **for**  $v \in \text{BG}$  **do**  
    Compute  $B = B_{\text{opt}}(S, \{v\})$ ;  
    **if**  $\text{min\_vol} > \text{Vol}(B)$  **then do**  
      Set  $\text{min\_vol} := \text{Vol}(B)$  and  $v^* := v$ ;  
    **od**  
  **end for**  
  Return  $B_{\text{opt}}(S, \{v^*\})$ ;  
**end** GRIDSEARCHMINVOLBBX

Figure 2: An exhaustive grid-based search algorithm for approximating  $B_{\text{opt}}(S)$

grid polyhedron  $P$ , and computed  $B_{\text{opt}}(P)$  exactly. We will next show that some grid point  $v^* \in \mathcal{G}$  is itself favorable in the sense that  $B_{\text{opt}}(S, \{ov^*\})$  is a  $(1 + \varepsilon)$ -approximation of  $B_{\text{opt}}(S)$ . Furthermore, the point  $v^*$  is close enough to the origin of  $\mathcal{G}$  so that we can perform an exhaustive search for this point. For this purpose we will compute  $\mathcal{CH}(S)$  explicitly and output  $B_{\text{opt}}(S, \{ov^*\})$  as the sought approximation.

The approximation algorithm is depicted in Figure 2. In a nutshell, the algorithm computes  $B^*(S)$ , builds the corresponding grid  $\mathcal{G}$ , and computes the box  $B_{\text{opt}}(S, \{v\})$  for all the grid points  $v \in \mathcal{G}$  close enough to  $o$ . The overall running time of the algorithm is  $O(n \log n + n/\varepsilon^3)$ . We next prove the correctness of the algorithm and (implicitly) compute the constant  $c$  which it uses.

Let  $B^*(S)$  be, as before, the bounding box of  $S$  computed by Lemma 3.6. Define the grid  $\mathcal{G} = \text{Grid}(c\varepsilon B^*(S))$ . The set of directions induced by  $\text{BG}$  (points computed by the algorithm) is a finite subset of  $\mathcal{G}$ . Let  $m = \max(\lceil 4/\varepsilon \rceil, 6)$ . Also, let  $B^{1/m}(S)$  be a translation of  $\frac{1}{m}B_{\text{opt}}(S)$  centered at  $o$ , and let  $\mathcal{B} = B_{\text{opt}}(S) \oplus B^{1/m}(S)$ . We assume without loss of generality that  $\mathcal{B}$  is axis parallel, its minimum in all axes is  $o$ , and that the longest edge of  $\mathcal{B}$  is parallel to the  $z$  axis. Set  $\delta = 1/(10m + 20)$ , and define a second grid  $\mathcal{M} = \text{Grid}(\delta\mathcal{B})$ . (The latter grid plays a role only in proving the correctness of the algorithm but not in the algorithm itself.) Lemma 3.6 tells us that the constant  $c$  may be chosen small enough so as to make the interior of every cell of  $\mathcal{M}$  contain at least one grid point of  $\mathcal{G}$ .

Is it fairly easy to prove that:

**Lemma 4.1** *Let  $R$  and  $H$  be a rectangle and a plane, respectively, in  $\mathbb{R}^3$ . There exists a rectangle  $R' \subset H$  that contains the orthogonal projection of  $R$  into  $H$ , such that  $\text{Area}(R') \leq \text{Area}(R)$ . ■*

Our next goal is to show that there exist two  $\mathcal{G}$ -grid points, where the minimum-volume bounding box of  $S$  perpendicular to the direction defined by these points is a good ap-

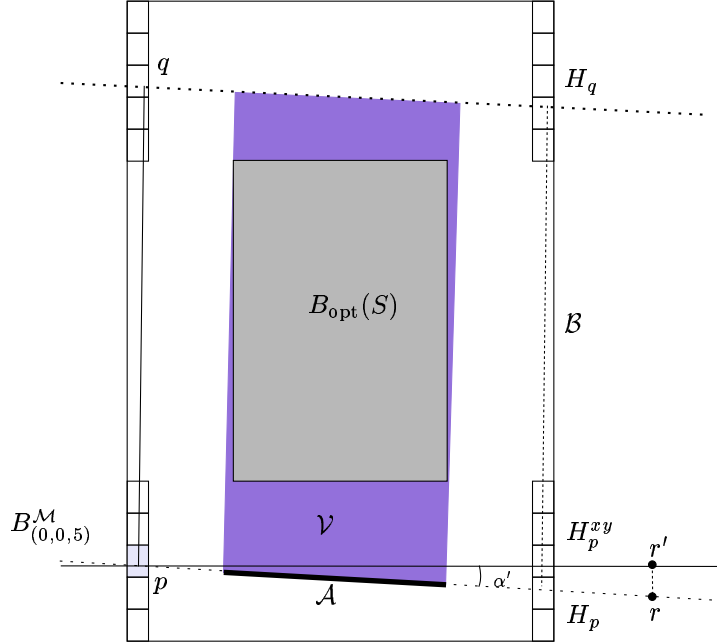


Figure 3: The construction used in Theorem 4.3

proximation of  $B_{\text{opt}}(S)$ . (Note that in the grid-search algorithm the first point is actually  $o$ .)

**Lemma 4.2** *Let  $p$  and  $q$  be points of  $\mathcal{G} \cap B_{(0,0,5)}^M$  and  $\mathcal{G} \cap B_{(0,0,20m+15)}^M$ , respectively. Then  $B_{\text{opt}}(S, \{pq\})$  is a  $(1 + \varepsilon)$ -approximation of  $B_{\text{opt}}(S)$ .*

*Proof:* Refer to Figure 3. Denote by  $H_p$  and  $H_q$  the two planes that pass through  $p$  and  $q$ , respectively, and are perpendicular to  $pq$ . Also denote the angle between  $pq$  and the  $z^+$  axis by  $\alpha$ . By construction, the direction  $pq$  is almost vertical: A simple calculation shows that  $\tan(\alpha) \leq \sqrt{2}/(10m)$ . Let  $H_p^{xy}$  be the  $xy$ -parallel plane that passes through  $p$ , let  $r \in H_p$  be an arbitrary point for which  $|pr| \leq 2|pq|$ , and let  $r'$  be the orthogonal projection of  $r$  into  $H_p^{xy}$ .

Then,

$$|rr'| \leq |pr| \tan(\alpha) \leq \frac{2\sqrt{2}|pq|}{10m} \leq \frac{3|pq|}{10m} \leq \frac{3|b_1|}{10m} \leq \frac{4|b_1|}{10m+20} = 4\delta|b_1|,$$

where we use the fact that  $3/(10m) \leq 4/(10m+20)$  if and only if  $m \geq 6$ , which is guaranteed. This implies that  $H_p$  cannot intersect  $B_{\text{opt}}(S)$  since the vertical distance between  $H_p$  and  $H_p^{xy}$  (for such a point  $r$ ) is less than  $4\delta|b_1|$  (four cells of  $M$ ), whereas the vertical distance between  $H_p^{xy}$  and the bottom of  $B_{\text{opt}}(S)$  is at least  $4\delta|b_1|$ . Similarly, the plane  $H_q$  cannot intersect the box  $B_{\text{opt}}(S)$ . The same argument shows that  $H_p$  and  $H_q$  intersect only the vertical edges of  $\mathcal{B}$  but do not intersect its top and bottom faces.

Let  $\mathcal{A}$  be the orthogonal projection of  $B_{\text{opt}}(S)$  into  $H_p$  (along  $pq$ ), and let  $\mathcal{V}$  be the product of  $\mathcal{A}$  and  $pq$  (that is, a prism whose base is  $\mathcal{A}$ ). It is easy to verify that  $B_{\text{opt}}(S) \subseteq \mathcal{V} \subseteq \mathcal{B}$ . Obviously, the rectangle  $R = \mathcal{B} \cap H_p^{xy}$  contains the set  $\mathcal{V} \cap H_p^{xy}$ . Moreover, according to

Lemma 4.1 the orthogonal projection of  $R$  into  $H_p$  is contained in a rectangle  $R' \in H_p$  such that  $\text{Area}(R') \leq \text{Area}(R)$ . Since  $\mathcal{V} \cap H_p^{xy} \subseteq R$ , we also have  $\mathcal{A} \subseteq R'$ . Let  $\hat{B}$  be the box with  $R'$  as its base and the opposite face lying on  $H_q$ . Clearly,  $S \subseteq B_{\text{opt}}(S) \subseteq \mathcal{V} \subseteq \hat{B}$ . Finally, we have

$$\text{Vol}(\hat{B}) = |pq| \text{Area}(R') \leq |b_1| \text{Area}(R) \leq \text{Vol}(\mathcal{B}) \leq (1 + \varepsilon) \text{Vol}(B_{\text{opt}}(S)).$$

Since  $\text{Vol}(\hat{B}) \geq \text{Vol}(B_{\text{opt}}(S, \{pq\}))$ , the latter box  $B_{\text{opt}}(S, \{pq\})$  is also a  $(1+\varepsilon)$ -approximation of  $B_{\text{opt}}(S)$ .  $\blacksquare$

We complete this discussion by showing that  $p\vec{q}$  is indeed “short,” namely, that  $|pq|_\infty$  (measured by  $\mathcal{G}$ -grid units) is small. Intuitively, this follows from the fact that the grid sizes of  $\mathbf{M}$  and  $\mathcal{G}$  are comparable up to a multiplicative constant. By Lemma 3.5 we have that a copy of the unit box of  $\mathbf{M}$  (that is,  $\delta\mathcal{B}$ ), scaled down by a constant factor and translated, is contained by the unit box of  $\mathcal{G}$  (that is,  $c\varepsilon B^*(S)$ ). In particular, this implies that every grid box of  $\mathbf{M}$  is covered by a constant number of grid boxes of  $\mathcal{G}$ . In addition, the segment  $pq$  is contained in  $\cup_{i=5}^{20m+15} B_{(0,0,i)}^{\mathbf{M}}$ , whose height (size along  $z$ ) is  $O(1/\varepsilon)$ . Thus, the segment  $pq$  can be covered by  $O(1/\varepsilon)$  grid boxes of  $\mathcal{G}$ . Hence, all the coordinates of  $q - p$  (in  $\mathcal{G}$  units) are  $O(1/\varepsilon)$ , where the constant of proportionality hidden in the big-Oh notation is  $c$  (the constant used by the algorithm). We thus establish the following theorem:

**Theorem 4.3** *Let  $S$  be a set of  $n$  points in  $\mathbb{R}^3$ , and let  $0 < \varepsilon \leq 1$  be a parameter. One can compute in  $O(n \log n + n/\varepsilon^3)$  time a bounding box  $B(S)$  with  $\text{Vol}(B(S)) \leq (1 + \varepsilon) \text{Vol}(B_{\text{opt}}(S))$ .*

The algorithm described above may be too slow to use in practice because the constant of proportionality hidden in the big-Oh notation (affected by the value of  $c$ ) may be too large. However, it suggests the heuristic of computing the bounding boxes  $B_{\text{opt}}(S, \{v\})$  induced by directions defined by grid points of  $\mathcal{G}$  “close” to  $o$ . Theorem 4.3 implies that the higher the bound on the “length” of  $v$  is, the better the approximation is.

## 5 Experimental Results

We have implemented software that computes the exact 2-dimensional minimum-area bounding rectangle of a planar point set (by computing its convex hull and then applying a rotating-calipers algorithm). Based on this tool we have implemented software that computes the exact 3-dimensional minimum-volume bounding box of a spatial point set, one of whose directions is given. We have used the latter tool for implementing several approximation heuristics for the minimum-volume bounding box, and report here on several of them. The entire software was implemented in plain C and it runs under any Unix-like operating system. The running times reported here were measured in a Linux environment on a 200-MHz Pentium-Pro machine. The software consists of about 1,500 lines of code.

It was easy to observe that for any bounding box  $B(S)$ , one can always “locally” improve (decrease) the volume of the box by projecting it into a plane perpendicular to one of the directions of  $B(S)$ , followed by computing the minimum-area bounding rectangle of the projected set in that plane, and by using this rectangle as the base of an improving

bounding box of  $S$ . Our experimental results revealed many examples in which this procedure converges to a local (but not a global) minimum-volume bounding box. Nevertheless, this procedure improves (by a small amount) the solutions produced by the approximation algorithms described in this paper. We performed each experiment twice, without and with this solution-improving step.

Here are three examples (out of the many examples we experimented with) of the performance of the heuristics. Figure 4(a.1) shows a rotated version of the set

$$S = \{(-1, -0.1, 0), (-1, 0.1, 0), (1, 0, -0.1), (1, 0, 0.1)\}.$$

(The points are displayed as small triangles.) Figures 4(a.2) and 4(a.3) show  $B^*(S)$  (which is also optimal among all boxes aligned with some diagonal of  $\mathcal{CH}(S)$ ), and the improved  $B^*(S)$ -G(20) (see below), respectively. Figure 4(b) shows a set  $S$  of 48 arbitrary points. The diameter of the set is shown as a nearly-vertical line segment. The figure shows  $B^*(S)$  (the nearly-vertical box) and the minimum-volume bounding box of  $S$  among those aligned with some hull diagonal. Figure 4(c) shows the same types of boxes bounding a set of 100 points which were randomly and uniformly selected on the unit sphere.

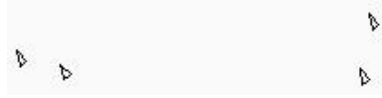
Table 1 shows the volumes of several bounding boxes of the three spatial sets, and the corresponding running times of our software. The box  $B^*(S)$  is the minimum-volume bounding box aligned with the diameter of the set  $S$ . The “all-pairs” box is obtained by minimizing the volume of all the boxes aligned with directions which connect some two points of  $S$ . The suffix “-G( $k$ )” stands for checking all the boxes aligned with directions obtained by connecting the origin  $o$  with a grid point whose  $L_\infty$  norm is at most  $k$ . The table reports results for Grid( $B^*(S)$ ) and for the regular Cartesian grid. The column entitled “MVBB( $v$ )” details the number of calls to the function that computes  $B_{\text{opt}}(S, \{v\})$ .

Note that although the more laborious heuristics require much time to run, some of the faster heuristics perform reasonably well in practice. For example, the  $B^*(S)$ -G(5) variant combined with the improvement step produces good approximating boxes and runs fast. Moreover, this heuristic performs considerably better than the uniform-grid heuristic for “long and skinny” sets of points.

## 6 Conclusion

In this paper we present an efficient algorithm for approximating the minimum-volume bounding box of a point set in  $\mathbb{R}^3$ . We also present a simpler algorithm which we implemented and experimented with on numerous three-dimensional point sets.

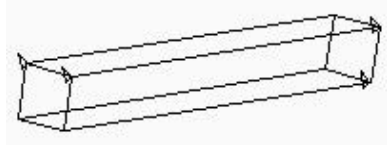
Jeff Erickson pointed out in a personal communication that it was possible to reduce the  $O(1/\varepsilon^{4.5})$  term in the running time of the first algorithm to  $O(1/\varepsilon^3)$  (in the cost of adding an  $O(n \log n)$  term) by using Dudley’s method [Dud74, AHSV97]. The main idea is to scale down the space so as to transform  $B^*(S)$  to a unit cube. There one computes a  $(c\varepsilon)$ -approximation (for a suitable constant  $c$ ) of  $\mathcal{CH}(S)$  and scales up this approximating polyhedron back to the original space. Similarly to our argumentation in Section 3, one can show that the minimum-volume bounding box of the scaled-up polyhedron is a  $(1 + \varepsilon)$ -approximation of the minimum-volume bounding box of  $S$ . This version of the algorithm requires  $O(n \log n + 1/\varepsilon^3)$  time.



(a.1) Points ( $S$ )

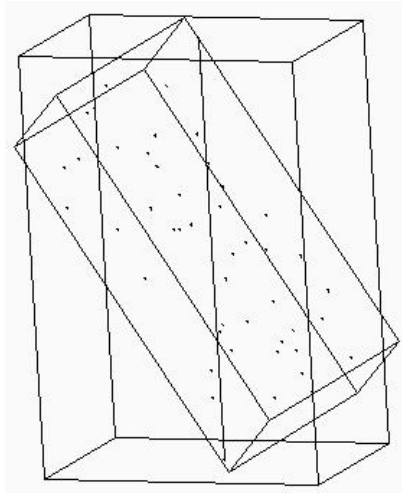


(a.3) Improved  $B^*(S)$ -G(20)

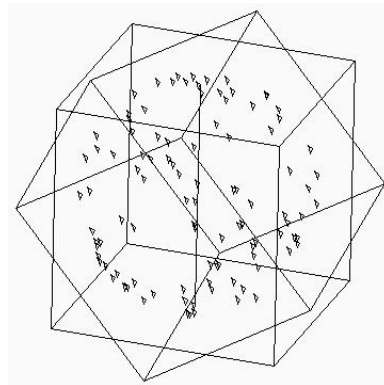


(a.2)  $B^*(S)$

(a) 4 points



(b) 48 points



(c) 100 points

Figure 4: Bounding boxes of three spatial point sets

S	Distribution	Box	Volume	Calls to MVBB (v)	Time	
					Per call	Total
4		$B^*(S)$	0.07980	1	(unreliable)	0
		All pairs	0.07980	1	(unreliable)	0
		$B^*(S)$ -G(2)	0.03995	23	(unreliable)	0
		(Improved)	0.03995	3	(unreliable)	0
		$B^*(S)$ -G(5)	0.03995	339	118 $\mu$ Sec	0.04 Sec
		(Improved)	0.03995	3	(unreliable)	0
		$B^*(S)$ -G(10)	0.03995	3107	113 $\mu$ Sec	0.35 Sec
		(Improved)	0.03995	3	(unreliable)	0
		$B^*(S)$ -G(20)	0.03995	26019	113 $\mu$ Sec	2.95 Sec
		(Improved)	0.03995	3	(unreliable)	0
		$xyz$ -G(2)	0.07974	23	(unreliable)	0
		(Improved)	0.05267	15	(unreliable)	0
		$xyz$ -G(5)	0.05674	339	118 $\mu$ Sec	0.04 Sec
		(Improved)	0.04202	27	(unreliable)	0.01 Sec
		$xyz$ -G(10)	0.05005	3107	119 $\mu$ Sec	0.37 Sec
		(Improved)	0.04009	9	(unreliable)	0
$xyz$ -G(20)	0.04082	26019	119 $\mu$ Sec	3.10 Sec		
(Improved)	0.04082	3	(unreliable)	0		
48	Arbitrary	$B^*(S)$	168.82	1	(unreliable)	0
		All pairs	83.20	1,128	674 $\mu$ Sec	0.76 Sec
		$B^*(S)$ -G(2)	87.11	23	(unreliable)	0.02 Sec
		(Improved)	83.24	18	(unreliable)	0.01
		$B^*(S)$ -G(5)	84.13	339	678 $\mu$ Sec	0.23 Sec
		(Improved)	83.39	12	(unreliable)	0
		$B^*(S)$ -G(10)	83.28	3,107	679 $\mu$ Sec	2.11 Sec
		(Improved)	83.18	9	(unreliable)	0
		$B^*(S)$ -G(20)	83.28	26,019	677 $\mu$ Sec	17.61 Sec
		(Improved)	83.18	9	(unreliable)	0.01 Sec
		$xyz$ -G(2)	83.22	23	(unreliable)	0.02 Sec
		(Improved)	83.20	6	(unreliable)	0
		$xyz$ -G(5)	83.22	339	678 $\mu$ Sec	0.23 Sec
		(Improved)	83.20	6	(unreliable)	0
		$xyz$ -G(10)	83.22	3,107	653 $\mu$ Sec	0.23 Sec
		(Improved)	83.20	6	(unreliable)	0.01 Sec
$xyz$ -G(20)	83.22	26,019	658 $\mu$ Sec	17.13 Sec		
(Improved)	83.11	6	(unreliable)	0		
100	Uniform on a unit sphere	$B^*$	7.333	1	(unreliable)	0
		All pairs	6.422	4,950	1,596 $\mu$ Sec	7.99 Sec
		$B^*(S)$ -G(2)	6.688	23	(unreliable)	0.04 Sec
		(Improved)	6.601	15	(unreliable)	0.02
		$B^*(S)$ -G(5)	6.446	339	1,622 $\mu$ Sec	0.55 Sec
		(Improved)	6.420	18	(unreliable)	0.03 Sec
		$B^*(S)$ -G(10)	6.427	3,107	1,641 $\mu$ Sec	5.10 Sec
		(Improved)	6.418	9	(unreliable)	0.02 Sec
		$B^*(S)$ -G(20)	6.421	26,019	1,625 $\mu$ Sec	42.27 Sec
		(Improved)	6.421	3	(unreliable)	0
		$xyz$ -G(2)	6.719	23	(unreliable)	0.04 Sec
		(Improved)	6.526	21	(unreliable)	0.03 Sec
		$xyz$ -G(5)	6.462	339	1,622 $\mu$ Sec	0.55 Sec
		(Improved)	6.418	12	(unreliable)	0.02 Sec
		$xyz$ -G(10)	6.440	3,107	1,629 $\mu$ Sec	5.06 Sec
		(Improved)	6.422	12	(unreliable)	0.02 Sec
$xyz$ -G(20)	6.426	26,019	1,638 $\mu$ Sec	42.63 Sec		
(Improved)	6.419	9	(unreliable)	0.01 Sec		

The implementation of the algorithm described in this paper is available online [Har00]. We conclude by mentioning two open problems:

- Can one maintain dynamically a  $(1 + \varepsilon)$ -approximation of the minimum-volume bounding box of a moving point set in  $\mathbb{R}^3$ ?
- Can one compute efficiently and by a simple algorithm a  $(1 + \varepsilon)$ -approximation of the minimum-volume bounding ellipsoid of a point set in  $\mathbb{R}^3$ ?

## Acknowledgement

The authors wish to thank Pankaj Agarwal, Jeff Erickson, and Micha Sharir for helpful discussions concerning the problem studied in this paper and related problems. This work is part of the second author's Ph.D. thesis, prepared at Tel-Aviv University under the supervision of Prof. Sharir.

## References

- [AHSV97] P.K. Agarwal, S. Har-Peled, M. Sharir, and K.R. Varadarajan. Approximate shortest paths on a convex polytope in three dimensions. *J. Assoc. Comput. Mach.*, 44:567–584, 1997.
- [And63] G. E. Andrews. A lower bound for the volume of strictly convex bodies with many boundary lattice points. *Trans. Amer. Math. Soc.*, 106:270–279, 1963.
- [BCG<sup>+</sup>96] G. Barequet, B. Chazelle, L.J. Guibas, J.S.B. Mitchell, and A. Tal. BOX-TREE: A hierarchical representation for surfaces in 3D. *Comput. Graph. Forum*, 15(3):C387–C396, C484, 1996. Proc. Eurographics'96.
- [BH99] G. Barequet and S. Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. In *Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, pages 82–91, 1999.
- [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.
- [BKSS90] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R\*-tree: An efficient and robust access method for points and rectangles. In *ACM SIGMOD Int. Conf. on Manag. of Data*, pages 322–331, 1990.
- [BS97] S. Bespamyatnikh and M. Segal. Covering a set of points by two axis-parallel boxes. In *Proc. 9th Canad. Conf. Comput. Geom.*, pages 33–38, 1997.
- [Cha96] T.M. Chan. Output-sensitive results on convex hulls, extreme points, and related problems. *Discrete Comput. Geom.*, 16:369–387, 1996.

- [CS89] K. L. Clarkson and P.W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.
- [Dud74] R.M. Dudley. Metric entropy of some classes of sets with differentiable boundaries. *J. Approx. Theory*, 10(3):227–236, 1974.
- [EK89] O. Egecioglu and B. Kalantari. Approximating the diameter of a set of points in the Euclidean space. *Inform. Process. Lett.*, 32:205–211, 1989.
- [FL95] C. Faloutsos and K. I. Lin. FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 24(2):163–174, June 1995.
- [FP87] O. D. Faugeras and J. Ponce. An object-centered hierarchical representation for 3-d objects: The prism-tree. *Computer Vision, Graphics, and Image Processing*, 38:1–28, 1987.
- [GK92] P. Gritzmann and V. Klee. Inner and outer  $j$ -radii of convex bodies in finite-dimensional normed spaces. *Discrete Comput. Geom.*, 7:255–280, 1992.
- [GLM96] S. Gottschalk, M.C. Lin, and D. Manocha. OBB-tree: A hierarchical structure for rapid interference detection. In *Proc. SIGGRAPH '96*, pages 171–180, 1996.
- [Har99] S. Har-Peled. Approximate shortest paths and geodesic diameters on convex polytopes in three dimensions. *Discrete Comput. Geom.*, 21:216–231, 1999.
- [Har00] S. Har-Peled. Source code of program for computing and approximating the diameter of a point-set in 3d, 2000. [http://www.uiuc.edu/~sariel/papers/00/diameter/diam\\_prog.html](http://www.uiuc.edu/~sariel/papers/00/diameter/diam_prog.html).
- [HKM95] M. Held, J.T. Klosowski, and J.S.B. Mitchell. Evaluation of collision detection methods for virtual reality fly-throughs. In *Proc. 7th Canad. Conf. Comput. Geom.*, pages 205–210, 1995.
- [Hub95] P. M. Hubbard. Collision detection for interactive graphics applications. *IEEE Trans. Visualization and Computer Graphics*, 1(3):218–230, September 1995.
- [O'R85] J. O'Rourke. Finding minimal enclosing boxes. *Internat. J. Comput. Inform. Sci.*, 14:183–199, 1985.
- [RL85] N. Roussopoulos and D. Leifker. Direct spatial search on pictorial databases using packed R-trees. *Proc. ACM SIGACT-SIGMOD Conf. Principles Database Systems*, pages 17–31, 1985.
- [Sam89] H. Samet. *Spatial Data Structures: Quadtrees, Octrees, and Other Hierarchical Methods*. Addison-Wesley, Reading, MA, 1989.
- [SRF87] T. Sellis, N. Roussopoulos, and C. Faloutsos. The  $R^+$ -tree: A dynamic index for multi-dimensional objects. In *Proc. 13th VLDB Conference*, pages 507–517, 1987.



- [Tou83] G. T. Toussaint. Solving geometric problems with the rotating calipers. In *Proc. IEEE MELECON '83*, pages A10.02/1–4, 1983.