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
E-mail: barequet@cs.technion.ac.il
and
Sariel Har-Peled
School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel
E-mail: sariel@math.tau.ac.il

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 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.

Additional Key Words and Phrases: approximation, bounding box

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 rectangular box that encloses S whose volume is at most $(1+\varepsilon)$ times the minimum volume of a bounding box of S.

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

Three-dimensional boxes that enclose sets of points are used for maintaining hierarchical partitioning of such sets. These data structures have important applications, e.g., in computer graphics (for fast rendering of a scene or for collision detection), and in statistics (for storing and performing range-search queries on a

---

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 [Barequet and Har-Peled 1999].
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.

large database of samples). From a top-down viewpoint, the problem in such applications is divided into two (admittedly, related) problems (a) of finding a good splitting of a given set of points into two (or more) subsets, e.g., in a way that roughly minimizes the maximum diameter of the subsets, and (b) of computing a nearly-optimal box (or another generic shape) that encloses each subset. Such a construction continues recursively and yields a hierarchical decomposition of the point set. In this paper we concentrate on the second problem.

Numerous heuristics have been proposed for computing a box that encloses a given set of points. The simplest heuristic is naturally that of computing the axis-aligned bounding box (AABB) of the point set. Two-dimensional variants of this heuristic yield hierarchical decompositions that include the well-known R-tree, the *packed* R-tree [Roussopoulos and Leifker 1985], the R^+-tree [Sellis et al. 1987], the R^*-tree [Beckmann et al. 1990], etc. Klosowski et al. [1998] use a minimum-volume AABB trimmed in a fixed number of directions for speeding up collision detection. Gottschalk et al. [1996] 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 Barequet et al. [1996] for their BOXTREE technique. 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—see also below). Other generic shapes, such as a sphere [Hubbard 1995], a cone [Samet 1989], or a prism [Faugeras and Ponce 1987; Barequet et al. 1996] 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 output with any guarantee (i.e., approximation factor of the optimal container).

An algorithm of Bespamyatnikh and Segal [1997] 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 [1985] presented the only algorithm (to the best of our knowledge) for computing the exact arbitrarily-oriented minimum-volume bounding box of a set S 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 in $\mathbb{R}^3$. Both algorithms use an approximation of the diameter of S to define a grid. The first algorithm approximates S by a set of grid points and computes the exact minimum-volume bounding box of the latter set. The second algorithm computes the box with a minimal volume out of a small number of bounding boxes whose orientations are induced by the grid. The first algorithm makes use of a result of Andrews [1963] that says that the number of vertices of the convex hull of a set of points in $\mathbb{R}^d$, with integer coordinates between 1 to k, is $O(k^{d(d-1)/(d+1)})$ (irrespective of the number of points). Thus we are able to achieve algorithms that 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 bound on the running time of the second algorithm is usually asymptotically weaker than that of the first algorithm, but the second algorithm is easier to implement.

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 algorithms. In Section 5 we present experimental results. We end in Section 6 with some concluding remarks.

2. NOTATIONS AND DEFINITIONS

We denote the origin of the coordinate frame by o and the unit cube by $\mathcal{C} = [0,1]^3$. 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 relevant set is assumed to contain n (or a comparable number of) points. We denote by $\mathrm{CH}(Q)$ (resp., $\mathrm{CH}(S)$) the 2-dimensional (resp., 3-dimensional) convex hull of Q (resp., S). The symbol P is used for denoting a convex polyhedron in 3-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 $\mathrm{Area}(R)$ and $\mathrm{Vol}(B)$ denote the area of R and the volume of B, respectively.

We denote by $R_{\mathrm{opt}}(Q)$ the *minimum-area bounding rectangle* of Q and by $B_{\mathrm{opt}}(S)$ the *minimum-volume bounding box* of S.[1] Let V be a set of orthogonal vectors in $\mathbb{R}^3$. We denote by $B_{\mathrm{opt}}(S, V)$ the minimum-volume bounding box of S whose set of directions contains V. We denote by $B^*(S)$ a bounding box of S whose volume is within a constant factor of the minimum volume of a bounding box of S (for some predefined positive constant). (Such a box is computed in Lemmas 3.3 and 3.5—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

$$\mathrm{Grid}(B) = \left\{ i_1 b_1 + i_2 b_2 + i_3 b_3 \;\middle|\; i_1, i_2, i_3 \in \mathbb{Z} \right\}.$$

Let $\mathsf{G} = \mathrm{Grid}(B)$. Denote by

$$B^{\mathsf{G}}_{(i,j,k)} = \left\{ x_1 b_1 + x_2 b_2 + x_3 b_3 \;\middle|\; \begin{array}{l} i \le x_1 \le i+1, \\ j \le x_2 \le j+1, \quad i,j,k \in \mathbb{Z} \\ k \le x_3 \le k+1, \end{array} \right\}$$

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

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

be the set of points of G whose L_∞-distance (along the directions b_1, b_2, b_3) from o is at most d.

[1] Since we are interested in the *area* (or *volume*) of the object, we may pick an arbitrary minimum bounding rectangle (or box) if several exist.

3. AN EFFICIENT APPROXIMATION ALGORITHM

In this section we present our main approximation algorithm for the problem of computing a 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.

DEFINITION 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 (where d is the dimension) of the diameter of S:[2]

LEMMA 3.1. *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 $|st| \le \mathcal{D}(S) \le \sqrt{d}|st|$.*

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 definition, $|st| \le \mathcal{D}(S)$, and clearly, $\mathcal{D}(S) \le \sqrt{d}\, l \le \sqrt{d}|st|$. 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 [Egecioglu and Kalantari 1989].)

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 [Har-Peled 1999] for a similar result), we include an easy proof of it here.

LEMMA 3.2. *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 $|st| \ge (1-\varepsilon)\mathcal{D}(S)$.*

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

$$|xy| - l \le |x_\mathsf{G} y_\mathsf{G}| \le |xy| + l.$$

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

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

The set S_G can be computed in $O(n)$ time (where the hidden constant of proportionality contains a factor d). The cardinality of S_G is $O(1/\varepsilon^d)$. Thus, we can compute the diameter of S_G (in a brute force manner) in $O(1/\varepsilon^{2d})$ time. The two

[2]Note that we approximate the diameter from *below*, while we approximate the minimum-volume bounding box from *above*.

points of S that correspond to the diameter of S_{G} are, by the above analysis, the sought $(1-\varepsilon)$-approximation of the diameter of S.

Any point of S_{G} that lies between two other points of S_{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_{G} that realize its diameter must be vertices of the convex-hull of S_{G}. Thus, the running time of the algorithm of Lemma 3.2 can be further improved by first computing the set of vertices of $\mathrm{CH}(S_{\mathsf{G}})$, denoted by $S_{\mathrm{CH}(S_{\mathsf{G}})}$. Set $h = |S_{\mathrm{CH}(S_{\mathsf{G}})}|$. Andrews [1963] showed that $h = O(1/\varepsilon^{d(d-1)/(d+1)})$. For $d \leq 3$ we compute $S_{\mathrm{CH}(S_{\mathsf{G}})}$ by computing the entire convex-hull of S_{G} in $O(|S_{\mathsf{G}}| \log |S_{\mathsf{G}}|)$ time. For higher dimensions we use an output-sensitive algorithm [Chan 1996]. 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 $S_{\mathrm{CH}(S_{\mathsf{G}})}$ (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 3.2 can be improved to $O\left(n + (1/\varepsilon)^{\frac{2d(d-1)}{d+1}}\right)$.

We can do even better in three dimensions if we are willing to sacrifice simplicity. In this case we compute the exact diameter of $\mathrm{CH}(S_{\mathsf{G}})$ in $O((1/\varepsilon^{3/2}) \log (1/\varepsilon))$ time [Clarkson and Shor 1989]. 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 $R_{\mathrm{opt}}(Q)$ can be done in $O(n \log n)$ time [Toussaint 1983]. (Hence, given a set S of n points and a direction v in $\mathbb{R}^3$, one can compute $B_{\mathrm{opt}}(S, \{v\})$ in $O(n \log n)$ time.) The bottleneck of the cited algorithm is the computation of $\mathrm{CH}(Q)$; when the latter is given in advance, $R_{\mathrm{opt}}(Q)$ can be computed in optimal $\Theta(n)$ time.

Lemma 3.3. *Given a set S of n points in $\mathbb{R}^3$, one can compute in $O(n)$ time a bounding box $B(S)$ with* $\mathrm{Vol}(B_{\mathrm{opt}}(S)) \leq \mathrm{Vol}(B(S)) \leq 6\sqrt{6}\,\mathrm{Vol}(B_{\mathrm{opt}}(S))$.

Proof. By using the algorithm of Lemma 3.1 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 3.1 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 μ be the direction perpendicular to st and $s't'$. We claim that the box $B^* = B_{\mathrm{opt}}(S, \{st, s't', \mu\})$ is a $(6\sqrt{6})$-approximation of $B_{\mathrm{opt}}(S)$.

Indeed, let R be the minimum-area bounding rectangle of Q in the directions $s't'$ and μ, let ω be the length of the edge of R in the direction μ, and let u, v be the

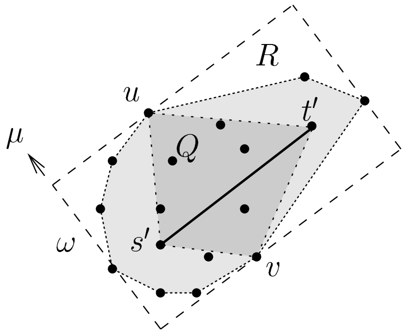


Fig. 1. The convex hull of the projection of S contains a large quadrangle

two points of Q lying on the two edges of R parallel to $s't'$. Clearly,

$$\mathrm{Area}(R) \le \omega \cdot \mathcal{D}(Q) \le \sqrt{2}\,\omega|s't'|.$$

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

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

We next show that $\mathrm{Vol}(\mathcal{S}) \ge \mathrm{Area}(\mathrm{CH}(Q))|st|/3$. For each $i = 1, \ldots, m$, the outer edge $p_i'q_i'$ of T_i is a projection of a pair of points $p_i, q_i \in S$ into H. Thus, each triangle T_i corresponds to a tetrahedron $\mathcal{T}_i = \mathrm{CH}(\{p_i, q_i, s, t\})$. These tetrahedra are pairwise disjoint in their interiors, and

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

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 is not incident with u (or, in more generality, parallel to the opposite face of u). We thus have

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

On the other hand,

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

Since $S \subseteq \mathcal{S} \subseteq B^*$, we have

$$\mathrm{Vol}(\mathcal{S}) \le \mathrm{Vol}(B_{\mathrm{opt}}(S)) \le \mathrm{Vol}(B^*) \le 6\sqrt{6}\,\mathrm{Vol}(\mathcal{S}),$$

as required. ∎

Note that the approximation factor for the diameter obtained in Lemma 3.3 can be improved by using better approximation schemes, e.g., the algorithm of

Lemma 3.2. However 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 3.3 can be extended to a d-dimensional space by choosing the direction of the diameter v of the point set S as one direction of the bounding box, projecting the points to a hyperplane H 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.[3] Moreover, the side lengths of the bounding box are found in decreasing order. Faloutsos and Lin [1995] use a similar method for visualizing a 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 convex polyhedron with sufficiently large volume contained in the unit cube $\mathcal{C}$ must contain an axis-parallel cube of sufficiently large size.

Lemma 3.4. *Let P be a convex polyhedron such that $P \subseteq \mathcal{C}$ and $\mathrm{Vol}(P) \geq 1/15$. 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, the width $\mathrm{Width}(P)$ of P satisfies

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

where $\mathrm{Width}(P)$ 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 [Gritzmann and Klee 1992] that $r(P) \geq \mathrm{Width}(P)/(2\sqrt{3})$. 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 $\mathbb{R}^3$. Recall that the *Minkowski sum* of A and B is the set $A \oplus B = \left\{ a + b \,\middle|\, a \in A, b \in B \right\}$. 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 $(1+c)^3 \mathrm{Vol}(B)$.

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

First we need a combined version of Lemmas 3.3 and 3.4:

Lemma 3.5. *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 $\mathrm{Vol}(B_{\mathrm{opt}}(S)) \leq \mathrm{Vol}(B^*(S)) \leq 15\,\mathrm{Vol}(B_{\mathrm{opt}}(S))$ and there exists a translation $v \in \mathbb{R}^3$ for which $\frac{1}{107}B^*(S) + v \subseteq \mathrm{CH}(S)$.*

[3] 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 $B_{\mathrm{opt}}(S)$. 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 $B_{\mathrm{opt}}(S)$.

PROOF. Let $B = B^*(S)$ be the bounding box of S computed by Lemma 3.3. 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 3.3 shows that $\mathrm{Vol}(T(\mathrm{CH}(S))) \geq \frac{1}{15}$ (we use here the fact that linear transformations preserve volume ratios). Hence by Lemma 3.4 there exists a translation v' such that

$$\frac{1}{107}(T(B) + t) + v' \subseteq T(\mathrm{CH}(S)) + t.$$

Therefore

$$\frac{1}{107}B + T^{-1}(v' - \frac{106}{107}t) \subseteq \mathrm{CH}(S),$$

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 3.3 degenerates to a line 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 $B_{\mathrm{opt}}(S)$.

Let $B = B^*(S)$ be the bounding box of S computed by Lemma 3.5, and let B_ε be a translated copy of $\frac{\varepsilon}{428}B$ centered at o. In addition, define $\mathcal{S}^\oplus = \mathrm{CH}(S) \oplus B_\varepsilon$ and $\mathsf{G} = \mathrm{Grid}(\frac{1}{2}B_\varepsilon)$. We approximate S on G. For each point $p \in S$ let $\mathsf{G}(p)$ be the set of eight vertices of the cell of G that contains p, and let $S_\mathsf{G} = \cup_{p \in S}\mathsf{G}(p)$. Define $P = \mathrm{CH}(S_\mathsf{G})$. Clearly, $\mathrm{CH}(S) \subseteq P \subseteq \mathcal{S}^\oplus$. 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 body is a box which contains at most $k = 860/\varepsilon + 4$ grid points along each of the directions of the edges of B, so k is also an upper bound for the number of grid points contained in any segment contained in P and parallel to any of these directions. Andrews [1963] 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 $B_{\mathrm{opt}}(P)$ exactly.

It remains to show that $B_{\mathrm{opt}}(P)$ is a $(1+\varepsilon)$-approximation of $B_{\mathrm{opt}}(S)$. Let $B^\varepsilon_{\mathrm{opt}}$ be a translation of $\frac{\varepsilon}{4}B_{\mathrm{opt}}(S)$ that contains B_ε. (The existence of $B^\varepsilon_{\mathrm{opt}}$ is guaranteed by Lemma 3.5.) Thus, $P \subseteq \mathrm{CH}(S) \oplus B_\varepsilon \subseteq \mathrm{CH}(S) \oplus B^\varepsilon_{\mathrm{opt}} \subseteq B_{\mathrm{opt}}(S) \oplus B^\varepsilon_{\mathrm{opt}}$. Since $B_{\mathrm{opt}}(S) \oplus B^\varepsilon_{\mathrm{opt}}$ is a box, it is a bounding box of P and therefore also of $\mathrm{CH}(S)$. Its volume is

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

as desired. (The last inequality is the only place where we use the assumption $0 \leq \varepsilon \leq 1$.)

The algorithm is depicted in Figure 2. Thus we obtain our main result:

THEOREM 3.6. *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 $B(S)$ of S with* $\mathrm{Vol}(B(S)) \leq (1+\varepsilon)\,\mathrm{Vol}(B_{\mathrm{opt}}(S))$. ■

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 with axes-directions parallel to those of $B(S)$ can be computed in additional $O(n)$ time.

ALGORITHM ApproxMinVolBbx (S, ε)
`Input:` A set S of n points in $\mathbb{R}^3$, and a parameter $0 < \varepsilon \leq 1$
`Output:` A $(1+\varepsilon)$-approximation of $B_{\rm opt}(S)$
begin
Compute the box $B^*(S)$ in $O(n)$ time; `/*` See Lemma 3.5 `*/`
Compute the point set S_{G} in $O(n)$ time;
Compute $P = \mathrm{CH}(S_{\mathsf{G}})$ 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_{G} along vertical lines of G. We have $O(1/\varepsilon^2)$ such points, thus computing their convex hull takes $O((1/\varepsilon^2)\log(1/\varepsilon))$ time. `*/`
Compute $B_{\rm opt}(P)$ `/*` using [O'Rourke 1985] `*/` in $O(1/\varepsilon^{4.5})$ time;
`/*` The complexity of P is $O(1/\varepsilon^{3/2})$ [Andrews 1963], so this step requires $O((1/\varepsilon^{3/2})^3) = O(1/\varepsilon^{4.5})$ time. `*/`
Return $B_{\rm opt}(P)$;
end ApproxMinVolBbx

Fig. 2. A fast algorithm for approximating $B_{\rm opt}(S)$

4. AN ALTERNATIVE PRACTICAL ALGORITHM

Unfortunately, in spite of its low asymptotic complexity, the algorithm described in the previous section is perhaps too difficult to implement, and the constant of proportionality in its running time bound is large. In this section we propose 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 $B^*(S) = B_{\rm opt}(S, \{v\})$, which served for the definition of the grid G. Then we expanded $\mathrm{CH}(S)$ (which was not computed explicitly) into a low-complexity grid polyhedron P, and computed $B_{\rm opt}(P)$ exactly. We will next show that some grid point $v^* \in \mathsf{G}$ is itself favorable in the sense that $B_{\rm opt}(S, \{ov^*\})$ is a $(1+\varepsilon)$-approximation of $B_{\rm opt}(S)$. Furthermore, the point v^* is close enough to the origin of G so that we can perform an exhaustive search for this point. For this purpose we will compute $\mathrm{CH}(S)$ explicitly and output $B_{\rm opt}(S, \{ov^*\})$ as the sought approximation.

The approximation algorithm is depicted in Figure 3. In a nutshell, the algorithm computes $B^*(S)$, builds the corresponding grid G, and computes the box $B_{\rm opt}(S, \{v\})$ for all the grid points $v \in \mathsf{G}$ 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 that it uses.

Let $B^*(S)$ be, as before, the bounding box of S computed by Lemma 3.5. Define the grid $\mathsf{G} = \mathrm{Grid}(c\varepsilon B^*(S))$. The set of directions induced by BG (the bounded grid of points computed by the algorithm) is a finite subset of the set of directions induced by G. Let $m = \max(\lceil 4/\varepsilon \rceil, 6)$. Also, let $B^{1/m}_{\rm opt}(S)$ be a translation of $\frac{1}{m}B_{\rm opt}(S)$ centered at o, and let $\mathcal{B} = B_{\rm opt}(S) \oplus B^{1/m}_{\rm opt}(S)$. We assume without loss of generality that $\mathcal{B}$ is axis parallel, that it has o as a vertex and lies in the positive octant, and that the longest edge b_1 of $\mathcal{B}$ is parallel to the z-axis. Set $\delta = 1/(10m + 20)$, and define a second grid $\mathsf{M} = \mathrm{Grid}(\delta\mathcal{B})$. (The latter grid plays a role only in proving the correctness of the algorithm but not in the algorithm

ALGORITHM GRIDSEARCHMINVOLBBX (S, ε)
```
    Input:  A set S of n points in R^3, and a parameter 0 < ε ≤ 1
    Output: A (1 + ε)-approximation of B_opt(S)
begin
    Compute CH(S);
    Compute B*(S); /* The box generated by Lemma 3.5 */
    Compute BG = G(B*(S), c/ε); /* Refer to the text for the value of c */
    Set min_vol := ∞ and v* := undefined;
    for v ∈ BG do
        Compute B = B_opt(S, {v});
        if min_vol > Vol(B) then do
            Set min_vol := Vol(B) and v* := v;
        end if
    end for
    Return B_opt(S, {v*});
end GRIDSEARCHMINVOLBBX
```

Fig. 3. An exhaustive grid-based search algorithm for approximating $B_{\text{opt}}(S)$

itself.) Lemma 3.5 tells us that the constant c may be chosen small enough so as to make the interior of every cell of M contain at least one grid point of G. (This is the only constraint on c. One can show that c can be chosen to be an absolute constant independent of ε—see also below.)

It is fairly easy to prove the following lemma.

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* $\text{Area}(R') \le \text{Area}(R)$.

Our next goal is to show that there exist two G-grid points, where the minimum-volume bounding box of S perpendicular to the direction defined by these points is a good approximation 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 $\mathsf{G} \cap B^{\mathsf{M}}_{(0,0,5)}$ and $\mathsf{G} \cap B^{\mathsf{M}}_{(0,0,20m+15)}$, respectively. Then $B_{\text{opt}}(S, \{pq\})$ is a $(1+\varepsilon)$-approximation of $B_{\text{opt}}(S)$.*

PROOF. Refer to Figure 4. 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 α. By construction, the direction pq is almost vertical: A simple calculation shows that $\tan(\alpha) \le \sqrt{2}/(10m)$. Let H_p^{xy} be the xy-parallel plane that passes through p, let $r \in H_p$ be an arbitrary point for which $|pr| \le 2|pq|$, and let r' be the orthogonal projection of r into H_p^{xy}.

Let α' denote the angle between rp and $r'p$. Since $\alpha' \le \alpha \le \pi/2$, we have

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

where we use the facts that $3/(10m) \le 4/(10m+20)$ if and only if $m \ge 6$, which is guaranteed, and that $|pq| \le |b_1|$, which follows from the choice of p and q. This

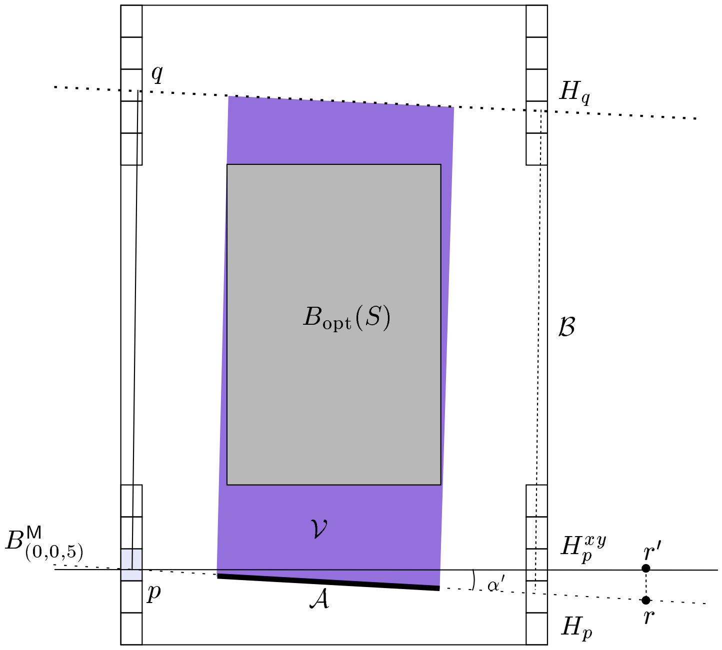


Fig. 4. The construction used in Theorem 4.3

implies that H_p cannot intersect $B_{\mathrm{opt}}(S)$ since the vertical distance between H_p and H_p^{xy} (for any point r that lies inside $\mathcal{B}$) is less than $4\delta|b_1|$ (four cells of M), whereas the vertical distance between H_p^{xy} and the bottom of $B_{\mathrm{opt}}(S)$ is at least $4\delta|b_1|$ (this follows from the definition of $\mathcal{B}$, the choice of p, and the value of δ). Similarly, the plane H_q cannot intersect the box $B_{\mathrm{opt}}(S)$. The same argument shows that H_p and H_q intersect only the vertical faces of $\mathcal{B}$ but do not intersect its top and bottom faces.

Let $\mathcal{A}$ be the orthogonal projection of $B_{\mathrm{opt}}(S)$ into H_p (along pq), and let $\mathcal{V}$ be the Cartesian product of $\mathcal{A}$ and pq (that is, a prism with base $\mathcal{A}$ and height $|pq|$). It is easy to verify that $B_{\mathrm{opt}}(S) \subseteq \mathcal{V} \subseteq \mathcal{B}$. 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 $\mathrm{Area}(R') \leq \mathrm{Area}(R)$. 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 with the opposite face lying on H_q. Clearly, $S \subseteq B_{\mathrm{opt}}(S) \subseteq \mathcal{V} \subseteq \hat{B}$. Finally, we have

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

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

We complete this discussion by showing that $\vec{pq}$ is indeed "short," namely, that $|pq|_\infty$ (measured by G-grid units) is small. Intuitively, this follows from the fact that the grid sizes of M and G are comparable up to a multiplicative constant. By Lemma 3.4 we have that a copy of the unit box of M (that is, $\delta\mathcal{B}$), scaled down by a constant factor and translated, is contained in the unit box of G (that is, in $c\varepsilon B^*(S)$). In particular, this implies that every grid box of M is covered by a constant number of grid boxes of G. In addition, the segment pq is contained in $\cup_{i=5}^{20m+15} B^{\mathsf{M}}_{(0,0,i)}$, whose height (size along the z-direction) is $O(1/\varepsilon)$. Thus, the segment pq can be covered by $O(1/\varepsilon)$ grid boxes of G. Hence, all the coordinates of $q - p$ (in 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. The algorithm* GRIDSEARCHMINVOLBBX *computes in $O(n \log n + n/\varepsilon^3)$ time a bounding box $B(S)$ of S with $\mathrm{Vol}(B(S)) \leq (1+\varepsilon)\,\mathrm{Vol}(B_{\mathrm{opt}}(S))$.*

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 $B_{\mathrm{opt}}(S, \{v\})$ induced by directions defined by grid points of 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

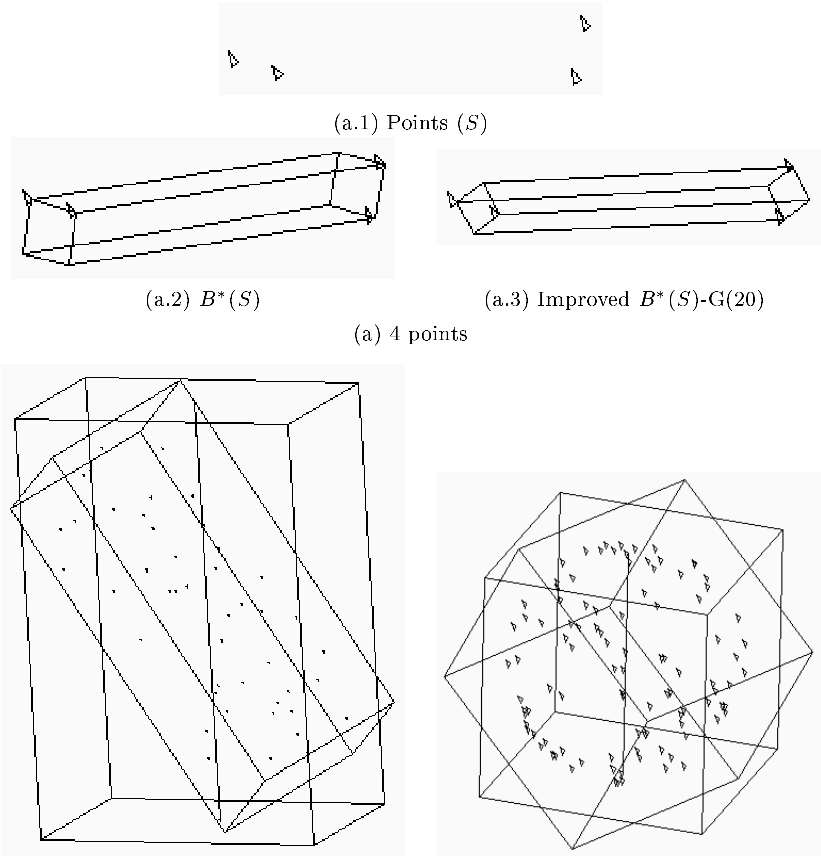
We have implemented software that computes the exact 2-dimensional minimum-area bounding rectangle of a planar point set (by computing its convex hull and then applying a rotating-calipers algorithm). Based on this tool we have implemented software that computes the exact 3-dimensional minimum-volume 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 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 5(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 5(a.2) and 5(a.3) show $B^*(S)$ (which is also optimal among all boxes aligned with some diagonal of $\mathrm{CH}(S)$), and the improved $B^*(S)$-G(20) (see below), respectively. Figure 5(b) shows a set S of 48 arbitrary points. The diameter of the set is shown as a nearly-vertical



(a.1) Points (S)

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

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

(a) 4 points

(b) 48 points

(c) 100 points

Fig. 5. Bounding boxes of three spatial point sets

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 5(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_∞ norm (relative to the grid) 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 a considerable amount of 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 pro-

$\lvert S\rvert$	Distribution	Box	Volume	Calls to MVBB(v)	Time Per call	Time 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 μSec	0.04 Sec
		(Improved)	0.03995	3	(unreliable)	0
		$B^*(S)$-G(10)	0.03995	3107	113 μSec	0.35 Sec
		(Improved)	0.03995	3	(unreliable)	0
		$B^*(S)$-G(20)	0.03995	26019	113 μSec	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 μSec	0.04 Sec
		(Improved)	0.04202	27	(unreliable)	0.01 Sec
		xyz-G(10)	0.05005	3107	119 μSec	0.37 Sec
		(Improved)	0.04009	9	(unreliable)	0
		xyz-G(20)	0.04082	26019	119 μSec	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 μSec	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 μSec	0.23 Sec
		(Improved)	83.39	12	(unreliable)	0
		$B^*(S)$-G(10)	83.28	3,107	679 μSec	2.11 Sec
		(Improved)	83.18	9	(unreliable)	0
		$B^*(S)$-G(20)	83.28	26,019	677 μSec	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 μSec	0.23 Sec
		(Improved)	83.20	6	(unreliable)	0
		xyz-G(10)	83.22	3,107	653 μSec	0.23 Sec
		(Improved)	83.20	6	(unreliable)	0.01 Sec
		xyz-G(20)	83.22	26,019	658 μSec	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 μSec	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 μSec	0.55 Sec
		(Improved)	6.420	18	(unreliable)	0.03 Sec
		$B^*(S)$-G(10)	6.427	3,107	1,641 μSec	5.10 Sec
		(Improved)	6.418	9	(unreliable)	0.02 Sec
		$B^*(S)$-G(20)	6.421	26,019	1,625 μSec	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 μSec	0.55 Sec
		(Improved)	6.418	12	(unreliable)	0.02 Sec
		xyz-G(10)	6.440	3,107	1,629 μSec	5.06 Sec
		(Improved)	6.422	12	(unreliable)	0.02 Sec
		xyz-G(20)	6.426	26,019	1,638 μSec	42.63 Sec
		(Improved)	6.419	9	(unreliable)	0.01 Sec

Table 1. Performance of the approximation heuristics

duces well approximating boxes and runs fast. Moreover, this heuristic performs considerably better than the uniform Cartesian-grid heuristic for "long and skinny" sets of points.

6. CONCLUSION

In this paper we presented an efficient algorithm for approximating the minimum-volume bounding box of a point set in $\mathbb{R}^3$. We have also presented 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 $O(1/\varepsilon^3)$ (at the cost of adding an $O(n \log n)$ term) by using Dudley's method [Dudley 1974; Agarwal et al. 1997]. 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 $\mathrm{CH}(S)$ and scales up this approximating polyhedron back to the original space. One can show, as in Section 3, 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.

We conclude by mentioning two open problems:

—Can one maintain dynamically and efficiently 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 $\mathbb{R}^3$?

ACKNOWLEDGMENTS

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

Agarwal, P. K., Har-Peled, S., Sharir, M., and Varadarajan, K. R. 1997. Approximate shortest paths on a convex polytope in three dimensions. *J. Assoc. Comput. Mach. 44*, 567–584.

Andrews, G. E. 1963. A lower bound for the volume of strictly convex bodies with many boundary lattice points. *Trans. of the American Mathematical Society 106*, 270–279.

Barequet, G., Chazelle, B., Guibas, L. J., Mitchell, J. S. B., and Tal, A. 1996. BOXTREE: A hierarchical representation for surfaces in 3D. *Computer Graphics Forum 15*, 3 (August), C387–C396, C484.

Barequet, G., and Har-Peled, S. 1999. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. *Proc. 10th Ann. ACM-SIAM Symp. on Discrete Algorithms*, 82–91.

Beckmann, N., Kriegel, H.-P., Schneider, R., and Seeger, B. 1990. The r*-tree: An efficient and robust access method for points and rectangles. *ACM SIGMOD Int. Conf. on Management of Data*, 322–331.

Bespamyatnikh, S., and Segal, M. 1997. Covering a set of points by two axis-parallel boxes. *Proc. 9th Canadian Conf. on Computational Geometry*, 33–38.

CHAN, T. M. 1996. Output-sensitive results on convex hulls, extreme points, and related problems. *Discrete & Computational Geometry 16*, 369–387.

CLARKSON, K. L., AND SHOR, P. W. 1989. Applications of random sampling in computational geometry, II. *Discrete & Computational Geometry 4*, 387–421.

DUDLEY, R. M. 1974. Metric entropy of some classes of sets with differentiable boundaries. *J. of Approximation Theory 10*, 3, 227–236.

EGECIOGLU, O., AND KALANTARI, B. 1989. Approximating the diameter of a set of points in the Euclidean space. *Information Processing Letters 32*, 205–211.

FALOUTSOS, C., AND LIN, K.-I. 1995. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. *ACM SIGMOD Record 24*, 2, 163–174.

FAUGERAS, O. D., AND PONCE, J. 1987. An object-centered hierarchical representation for 3-d objects: The prism-tree. *Computer Vision, Graphics, and Image Processing 38*, 1–28.

GRITZMANN, P., AND KLEE, V. 1992. Inner and outer j-radii of convex bodies in finite-dimensional normed spaces. *Discrete & Computational Geometry 7*, 255–280.

GOTTSCHALK, S., LIN, M. C., AND MANOCHA, D. 1992. OBB-tree: A hierarchical structure for rapid interference detection. *Proc. SIGGRAPH '96*, 171–180.

HAR-PELED, S. 1999. Approximate shortest paths and geodesic diameters on convex polytopes in three dimensions. *Discrete & Computational Geometry 21*, 216–231.

HUBBARD, P. M. 1995. Collision detection for interactive graphics applications. *IEEE Trans. on Visualization and Computer Graphics 1*, 3 (September), 218–230.

KLOSOWSKI, J., HELD, M., MITCHELL, J. S. B., ZIKAN, K., AND SOWIZRAL, H. 1998. Efficient collision detection using bounding volume hierarchies of k-DOPs. *IEEE Trans. on Visualization and Computer Graphics 4*, 1, 21–36.

O'ROURKE, J. 1985. Finding minimal enclosing boxes. *Int. J. Comput. Inform. Sci. 14*, June, 183–199.

ROUSSOPOULOS, N., AND LEIFKER, D. 1985. Direct spatial search on pictorial databases using packed r-trees. *Proc. ACM SIGACT-SIGMOD Conf. on Principles of Database Systems*, 17–31.

SAMET, H. 1989. *Spatial Data Structures: Quadtrees, Octrees, and Other Hierarchical Methods.* Addison-Wesley, Reading, MA,

SELLIS, T., ROUSSOPOULOS, N., AND FALOUTSOS, C. 1987. The R$^+$-tree: A dynamic index for multi-dimensional objects. *Proc. 13th Conf. on Very Large Databases*, 507–517.

TOUSSAINT, G. T. 1983. Solving geometric problems with the rotating calipers. *Proc. IEEE MELECON*, 1–4.