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

Author: Gill Barequet and Sariel Har-Peled
Presenter: An Nguyen
The Problem

- Input: a set $S$ of points in $\mathbb{R}^3$, a parameter $0 < \varepsilon < 1$
- Output: A bounding box enclosing $S$ and approximating the minimum bounding box of $S$ by a factor $(1+\varepsilon)$
Related Works

- Axis-aligned bounding box (AABB)
- Oriented bounding box (OBB)

Computations take $O(n)$ time and space
Heuristics, no quality guarantee!
Related Works

- Exact algorithm to compute minimum-volume bounding box:
  - $O(n \log n)$ time in $R^2$, $O(n)$ if $CH(R)$ is known.
  - $O(n^3)$ time in $R^3$
Notations

- **Box** \( B = (b_1, b_2, b_3) \)
- **Grid points** \( \text{Grid}(B) = \left\{ i_1 b_1 + i_2 b_2 + i_3 b_3 \mid i_1, i_2, i_3 \in \mathbb{Z} \right\} \)
- **Cell**

\[
\begin{align*}
B_{(i,j,k)}^G &= \left\{ x_1 b_1 + x_2 b_2 + x_3 b_3 \mid 
\begin{array}{l}
  i \leq x_1 \leq i + 1, \\
  j \leq x_2 \leq j + 1, \\
  k \leq x_3 \leq k + 1,
\end{array}
, i, j, k \in \mathbb{Z} \right\}
\end{align*}
\]
Coarse Approximation for Diameter

Lemma: A pair of points \((s,t)\) such that \(|st| \leq \mathcal{D}(S) \leq \sqrt{d} \ |st|\) can be computed in \(O(n)\) time.

In particular, we have a linear time approximate algorithm

- \((1/\sqrt{2})\)-approximate algorithm in \(\mathbb{R}^2\)
- \((1/\sqrt{3})\)-approximate algorithm in \(\mathbb{R}^3\)
Lemma: A pair of points \((s, t)\) such that \(|st| \geq (1-\varepsilon)\mathcal{D}(S)\) can be computed in time \(O(n + 1/\varepsilon^{2(d-1)})\)

1. original points \(S\) compute \(B(S)\)
2. snap \(S\) to grid\((\varepsilon/(2\sqrt{d}) B)\) to get \(S_g\)
3. keep only extreme grid points brute force to find the diameter \((s_g, t_g)\) and thus \((s, t)\)
(1+\(\varepsilon\))-Approximate Algorithm for Diameter

Lemma: A pair of points (s, t) such that \(|st| \geq (1-\varepsilon)D(S)\) can be computed in time \(O(n+1/\varepsilon^{2(d-1)})\)

- original points S compute \(B(S)\) cost: \(O(n)\)
- snap S to grid(\(\varepsilon/(2\sqrt{d})B\)) grid size \(\sim O(\varepsilon)\), \(|S_G| = O(1/\varepsilon^d)\) cost: \(O(n)\)
- keep only extreme grid points \(\Rightarrow O(1/\varepsilon^{d-1})\) remains brute force to find the diameter cost: \(O(1/\varepsilon^{2(d-1)})\)
(1+\(\varepsilon\))-Approximate Algorithm for Diameter

Lemma: A pair of points \((s,t)\) such that \(|st| \geq (1-\varepsilon)D(S)\) can be computed in time \(O(n+1/\varepsilon^{2(d-1)})\)

\[|st| \geq |s_G t_G| - \mathcal{L} = D(S_G) - \mathcal{L} \geq D(S) - 2\mathcal{L} \geq (1-\varepsilon) D(S)\]
Can we improve?

- Lemma: A pair of points \((s,t)\) such that \(|st| \geq (1-\epsilon)D(S)\) can be computed in time \(O(n+1/\epsilon^{2(d-1)})\)

original points \(S\) compute \(B(S)\)

snap \(S\) to grid\((\epsilon/(2\sqrt{d}) B)\) to get \(S_G\)

keep only extreme grid points brute force to find the diameter \((s_G, t_G)\) and thus \((s,t)\)
Computing Extreme Points

- \( \mathcal{CH}(S_g) \) has size \( h = |\mathcal{CH}(S_g)| = O(1/\varepsilon^{(d-1)d/(d+1)}) \)
- \( S_g \) has size \( m = O(1/\varepsilon^{d-1}) \)

- \( \mathcal{CH}(S_g) \) can be computed (using output sensitive alg.)
  \[
  O(m \log^{d+2} h + (mh)^{1-1/(d/2 + 1)} \log^{O(1)} m) \\
  = O(1/\varepsilon^{2(d-1)d/(d+1)})
  \]

- Brute force computation of diameter:
  \[
  O(h^2) = O(1/\varepsilon^{2(d-1)d/(d+1)})
  \]

\( \Rightarrow \) \( O(n + 1/\varepsilon^{2(d-1)d/(d+1)}) \) time algorithm

\( O(n + 1/\varepsilon^3) \) in \( \mathbb{R}^3 \)
Better Diameter Computation in \( \mathbb{R}^3 \)

- Exact diameter of \( CH(S_g) \) can be computed in time \( O((1/\varepsilon^{3/2}) \log(1/\varepsilon)) \)

\( \Rightarrow \) \( O(n + (1/\varepsilon^{3/2}) \log(1/\varepsilon)) \) time algorithm
Coarse Approximation Algorithm for Bounding Box

Lemma: we can compute in $O(n)$ time a bounding box $B(S)$ such that
$\text{Vol}(B_{\text{opt}}(S)) \leq \text{Vol}(B(S)) \leq 6\sqrt{6} \text{Vol}(B_{\text{opt}}(S))$

Idea (in 2D, for point set $Q$)
1. Compute approximate diameter $s't'$
2. Compute rectangle $R$ with direction $s't'$
3. Observe:
   $|s't'| \leq |xy| \leq D(Q) \leq |st|/\sqrt{2}$

the width of $R$

$\text{Area}(R(Q)) = 2 \text{Area}(s'bt'a) \leq 2\sqrt{2} \text{Area}(sbta) \leq 2\sqrt{2} \mathcal{CH}(Q) \leq 2\sqrt{2} \text{Area}(R_{\text{opt}}(Q))$
Coarse Box Property

Lemma: There exists a translation $v \in \mathbb{R}^3$ for which $(1/107)B(S) + v \subseteq \mathcal{CH}(S)$

Proof idea:
- Suppose $B(S)$ is an axis-aligned unit cube
- Vol($\mathcal{CH}(S)$) $\geq 1/(6\sqrt{6})$
- A unit cube has diameter $\sqrt{3}$, any cross section has area $\leq 3\pi/4$
- $\mathcal{CH}(S)$ has width $\geq 2/(9\sqrt{6\pi})$
- $\mathcal{CH}(S)$ inscribes a ball of radius $2/(9\sqrt{6\pi})/(2\sqrt{3}) = 1/(27\sqrt{2\pi})$
- $\mathcal{CH}(S)$ inscribes an axis-aligned cube of size $(2/\sqrt{3})/(27\sqrt{2\pi}) = \sqrt{2} / (27\sqrt{3\pi}) > 1/104$
(1+\(\varepsilon\))-Approximate Algorithm for Diameter (revisit)

- Lemma: A pair of points \((s,t)\) such that \(|st| \geq (1-\varepsilon)D(S)\) can be computed in time \(O(n+1/\varepsilon^{2(d-1)})\)

- original points \(S\)
- compute \(B(S)\)
- \(\geq D(S)\)
- snap \(S\) to grid\((\varepsilon/(2\sqrt{d})\) B) to get \(S_g\)
- keep only extreme grid points
- brute force to find the diameter \((s_g, t_g)\) and thus \((s,t)\)
(1+\(\varepsilon\))-Approximate Algorithm for Bounding Box

- Thm: A box \(B(S)\) such that \(B(S) \leq (1+\varepsilon) B_{opt} S\) can be computed in time \(O(n+1/\varepsilon^{4.5})\)

- original points \(S\) compute coarse box \(B\)
- expand \(S\) to grid(\(\varepsilon/428\) B) to get \(S_{G}\)
- keep only extreme grid points and their convex hull
- brute force to find the bounding box \(B^\varepsilon_{opt}\)
(1+\(\varepsilon\))-Approximate Algorithm for Bounding Box

- **Thm**: A box \(B(S)\) such that \(B(S) \leq (1+\varepsilon) B_{\text{opt}}S\) can be computed in time \(O(n+1/\varepsilon^{4.5})\)

- **original points** \(S\) compute coarse box \(B\)
  
  Cost: \(O(n)\)

- **expand** \(S\) to grid(\(\varepsilon/428 B\)) to get \(S_\varepsilon\)
  
  Cost: \(O(n)\)

- **Convex hull**
  
  Cost: \(O(1/\varepsilon^2 \log(1/\varepsilon))\)
  Output: \(O(1/\varepsilon^{3/2})\) points

- **Compute**: \(B^\varepsilon_{\text{opt}}\)
  
  Cost: \(O(1/(\varepsilon^{3/2})^3)\)
(1+\(\varepsilon\))-Approximate Algorithm for Bounding Box

- Thm: A box \(B(S)\) such that \(B(S) \leq (1+\varepsilon) B_{\text{opt}}(S)\) can be computed in time \(O(n+1/\varepsilon^{4.5})\)

- Let \(B^\varepsilon = (1/428)B, B^\varepsilon_{\text{opt}} = \varepsilon/4 B_{\text{opt}}(S)\) such that \(B^\varepsilon_{\text{opt}}\) contains \(B^\varepsilon\).

- \(P \subset CH(S) \oplus B^\varepsilon\)
  \(\subset CH(S) \oplus B^\varepsilon_{\text{opt}}\)
  \(\subset B_{\text{opt}}(S) \oplus B^\varepsilon_{\text{opt}}\)
  \(= (1+\varepsilon/4) B_{\text{opt}}(S)\)

- \(\text{Vol}(B_{\text{opt}}(P)) \leq (1+\varepsilon/4)^3 \text{Vol}(B_{\text{opt}}(S)) \leq (1+\varepsilon) B_{\text{opt}}(S)\)

We don’t know \(B_{\text{opt}}(S)\), yet we know \(B^\varepsilon_{\text{opt}}\) exists!
Algorithm is too complicated to implement!
Grid Search Algorithm

- Thm: A box $B(S)$ such that $B(S) \leq (1+\epsilon) B_{opt}(S)$ can be computed in time $O(n \log n + n/\epsilon^3)$

Idea: if the direction $v$ of a side of $B_{opt}(S)$ is [approximately] known, we can project $S$ to some plane $H$ perpendicular to $v$ then compute the optimal rectangle bounding of the projected points.

![Diagram showing grid search algorithm with plane H and direction v]
The Algorithm

**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 \(B_{opt}(S)\).

begin

Compute \(CH(S)\);

Compute \(B^*(S)\); /* The box generated by Lemma 3.6 */

Compute \(BG = G(B^*(S), c/\varepsilon)\); /* Refer to the text for the value of \(c\) */

Set \(\text{min}_\text{vol} := \infty\) and \(v^* := \text{undefined}\);

for \(v \in BG\) do

Compute \(B = B_{opt}(S, \{v\})\);

if \(\text{min}_\text{vol} > \text{Vol}(B)\) then do

Set \(\text{min}_\text{vol} := \text{Vol}(B)\) and \(v^* := v\);

od

end for

Return \(B_{opt}(S, \{v^*\})\);

end GridSearchMinVolBbx
Proof of Correctness

- If we generate enough directions, one direction $v$ will be close enough to the direction of the longest edge of the optimal bounding box $B_{\text{opt}}$

- The optimal box with direction $v$ containing $B_{\text{opt}}$ has volume only slightly larger than $B_{\text{opt}}$
Algorithm is too slow for implementation!
Implementation

- Algorithms:
  1. Compute $B^*$ (box along approximate diagonals)
  2. Find $B(S,v)$ for certain direction $v$ (grid search algorithm)
  3. Local refinement (re-projection, recompute optimal rectangle)

- Input
  - 4 points
  - 48 random points
  - 100 random points on a sphere

Optimal solution is not computed for comparison!
## Results – 4 points

| $|S|$ | Distribution | Box | Volume | Calls to MVBB(v) | Time Per call | Total |
|-----|--------------|-----|--------|-----------------|---------------|--------|
| 4   | 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 |

- along appx diagonal
- min-vol along edges
- grid search, along B*
- search size
- reprojection
- grid search, std. coord.
## Results – 48 points

<table>
<thead>
<tr>
<th>48</th>
<th>Arbitrary</th>
<th>$B^*(S)$</th>
<th>168.82</th>
<th>1</th>
<th>(unreliable)</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>All pairs</td>
<td>83.20</td>
<td>1,128</td>
<td>674 μSec</td>
<td>0.76 Sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$B^*(S)$-G(2)</td>
<td>87.11</td>
<td>23</td>
<td>(unreliable)</td>
<td>0.02 Sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Improved)</td>
<td>83.24</td>
<td>18</td>
<td>(unreliable)</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$B^*(S)$-G(5)</td>
<td>84.13</td>
<td>339</td>
<td>678 μSec</td>
<td>0.23 Sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Improved)</td>
<td>83.39</td>
<td>12</td>
<td>(unreliable)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$B^*(S)$-G(10)</td>
<td>83.28</td>
<td>3,107</td>
<td>679 μSec</td>
<td>2.11 Sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Improved)</td>
<td>83.18</td>
<td>9</td>
<td>(unreliable)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$B^*(S)$-G(20)</td>
<td>83.28</td>
<td>26,019</td>
<td>677 μSec</td>
<td>17.61 Sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Improved)</td>
<td>83.18</td>
<td>9</td>
<td>(unreliable)</td>
<td>0.01 Sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$xyz$-G(2)</td>
<td>83.22</td>
<td>23</td>
<td>(unreliable)</td>
<td>0.02 Sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Improved)</td>
<td>83.20</td>
<td>6</td>
<td>(unreliable)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$xyz$-G(5)</td>
<td>83.22</td>
<td>339</td>
<td>678 μSec</td>
<td>0.23 Sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Improved)</td>
<td>83.20</td>
<td>6</td>
<td>(unreliable)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$xyz$-G(10)</td>
<td>83.22</td>
<td>3,107</td>
<td>653 μSec</td>
<td>0.23 Sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Improved)</td>
<td>83.20</td>
<td>6</td>
<td>(unreliable)</td>
<td>0.01 Sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$xyz$-G(20)</td>
<td>83.22</td>
<td>26,019</td>
<td>658 μSec</td>
<td>17.13 Sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Improved)</td>
<td>83.11</td>
<td>6</td>
<td>(unreliable)</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>Uniform on a unit sphere</td>
<td>$B^*$</td>
<td>7.333</td>
<td>1</td>
<td>(unreliable)</td>
<td>0</td>
</tr>
<tr>
<td>-----</td>
<td>--------------------------</td>
<td>-------</td>
<td>-------</td>
<td>---</td>
<td>--------------</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>All pairs</td>
<td>6.422</td>
<td>4,950</td>
<td>1,596 μSec</td>
<td>7.99 Sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$B^*(S)$-G(2)</td>
<td>6.688</td>
<td>23</td>
<td>(unreliable)</td>
<td>0.04 Sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Improved)</td>
<td>6.601</td>
<td>15</td>
<td>(unreliable)</td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$B^*(S)$-G(5)</td>
<td>6.446</td>
<td>339</td>
<td>1,622 μSec</td>
<td>0.55 Sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Improved)</td>
<td>6.420</td>
<td>18</td>
<td>(unreliable)</td>
<td>0.03 Sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$B^*(S)$-G(10)</td>
<td>6.427</td>
<td>3,107</td>
<td>1,641 μSec</td>
<td>5.10 Sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Improved)</td>
<td>6.418</td>
<td>9</td>
<td>(unreliable)</td>
<td>0.02 Sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$B^*(S)$-G(20)</td>
<td>6.421</td>
<td>26,019</td>
<td>1,625 μSec</td>
<td>42.27 Sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Improved)</td>
<td>6.421</td>
<td>3</td>
<td>(unreliable)</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$xyz$-G(2)</td>
<td>6.719</td>
<td>23</td>
<td>(unreliable)</td>
<td>0.04 Sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Improved)</td>
<td>6.526</td>
<td>21</td>
<td>(unreliable)</td>
<td>0.03 Sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$xyz$-G(5)</td>
<td>6.462</td>
<td>339</td>
<td>1,622 μSec</td>
<td>0.55 Sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Improved)</td>
<td>6.418</td>
<td>12</td>
<td>(unreliable)</td>
<td>0.02 Sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$xyz$-G(10)</td>
<td>6.440</td>
<td>3,107</td>
<td>1,629 μSec</td>
<td>5.06 Sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Improved)</td>
<td>6.422</td>
<td>12</td>
<td>(unreliable)</td>
<td>0.02 Sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$xyz$-G(20)</td>
<td>6.426</td>
<td>26,019</td>
<td>1,638 μSec</td>
<td>42.63 Sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Improved)</td>
<td>6.419</td>
<td>9</td>
<td>(unreliable)</td>
<td>0.01 Sec</td>
<td></td>
</tr>
</tbody>
</table>
Summary

- Two algorithms to compute $(1+\varepsilon)$-approximation of the minimum volume bounding box
  - $O(n + 1/\varepsilon^{4.5})$
  - $O(n \log n + 1/\varepsilon^{3})$
- Heuristics for more practical algorithms