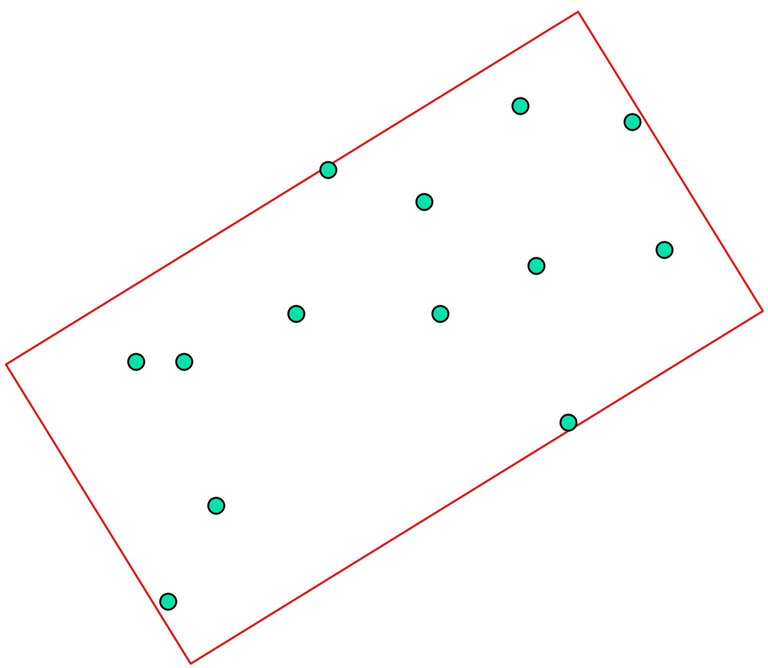
Efficiently Approximating the Minimum-Volume Bounding Box of a Point Set in Three Dimensions

Author: Gill Barequet and Sariel Har-Peled

Presenter: An Nguyen

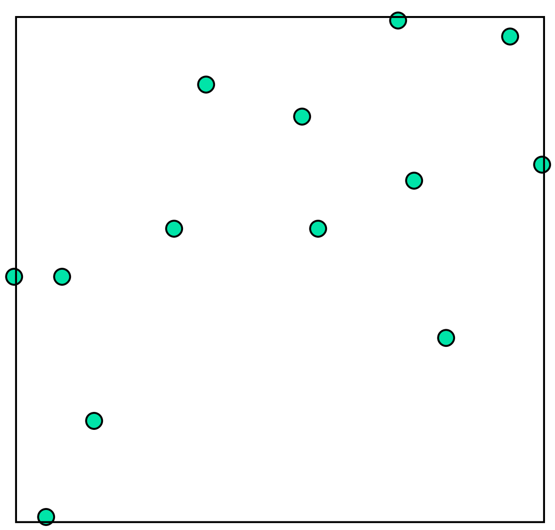
The Problem

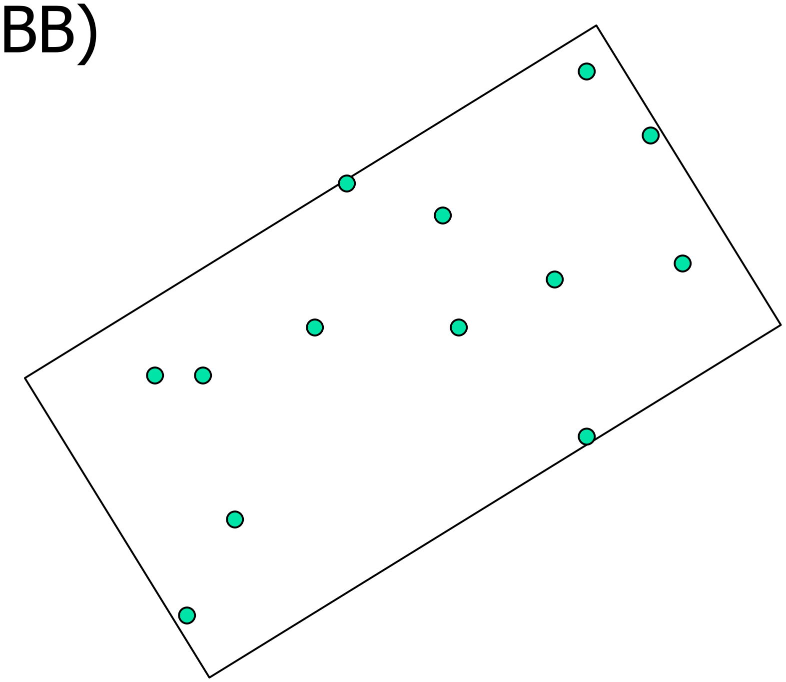
- Input: a set S of points in R^3, a parameter $0 < \varepsilon < 1$
- Output: A bounding box enclosing S and approximating the minimum bounding box of S by a factor $(1+\varepsilon)$



Related Works

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





Computations take O(n) time and space

Heuristics, no quality guarantee!

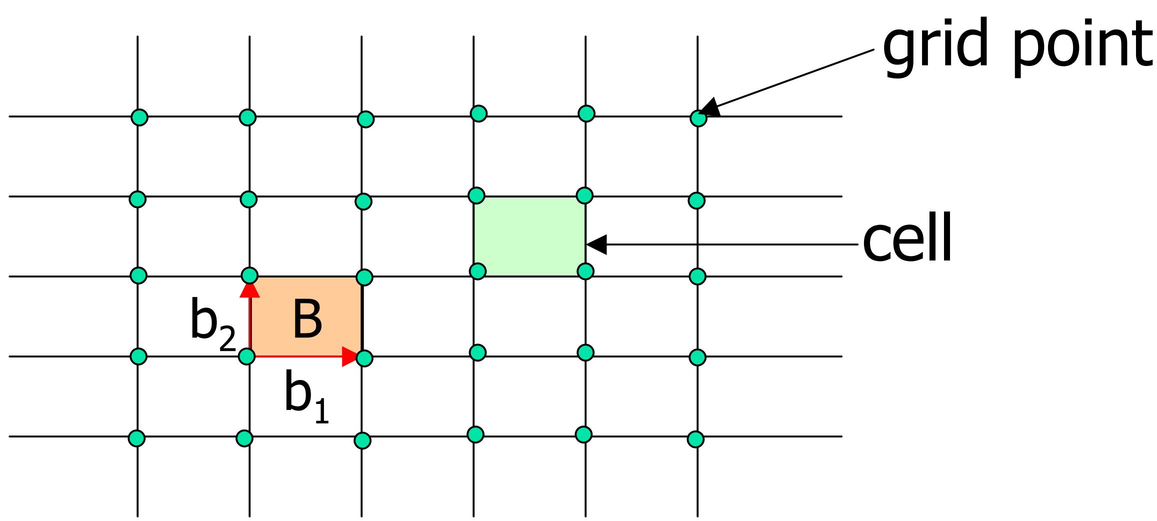
Related Works

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

Notations

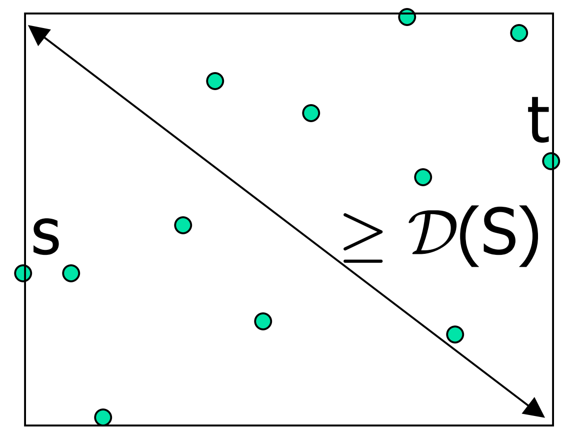
- Box $B = (b_1, b_2, b_3)$
- Grid points $\mathrm{Grid}(B) = \left\{ i_1 b_1 + i_2 b_2 + i_3 b_3 \;\middle|\; i_1, i_2, i_3 \in \mathbb{Z} \right\}$
- Cell

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



Coarse Approximation for Diameter

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

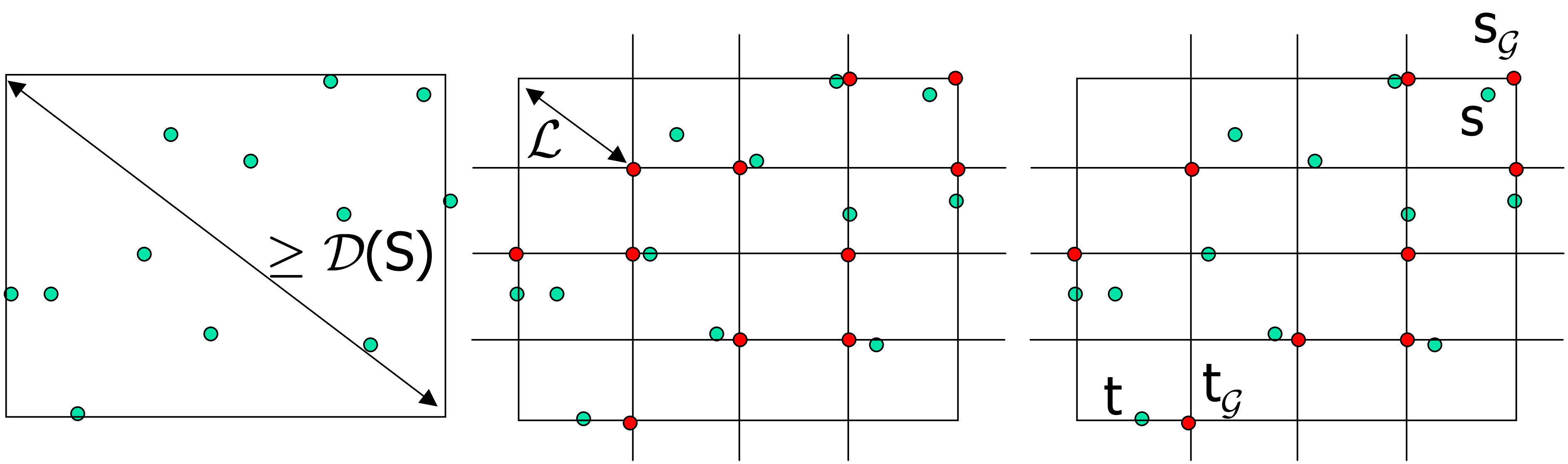


$\mathcal{D}(S)$: diameter of S

In particular, we have a linear time approximate algorithm
$(1/\sqrt{2})$-approximate algorithm in R^2
$(1/\sqrt{3})$-approximate algorithm in R^3

$(1+\varepsilon)$-Approximate Algorithm for Diameter

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



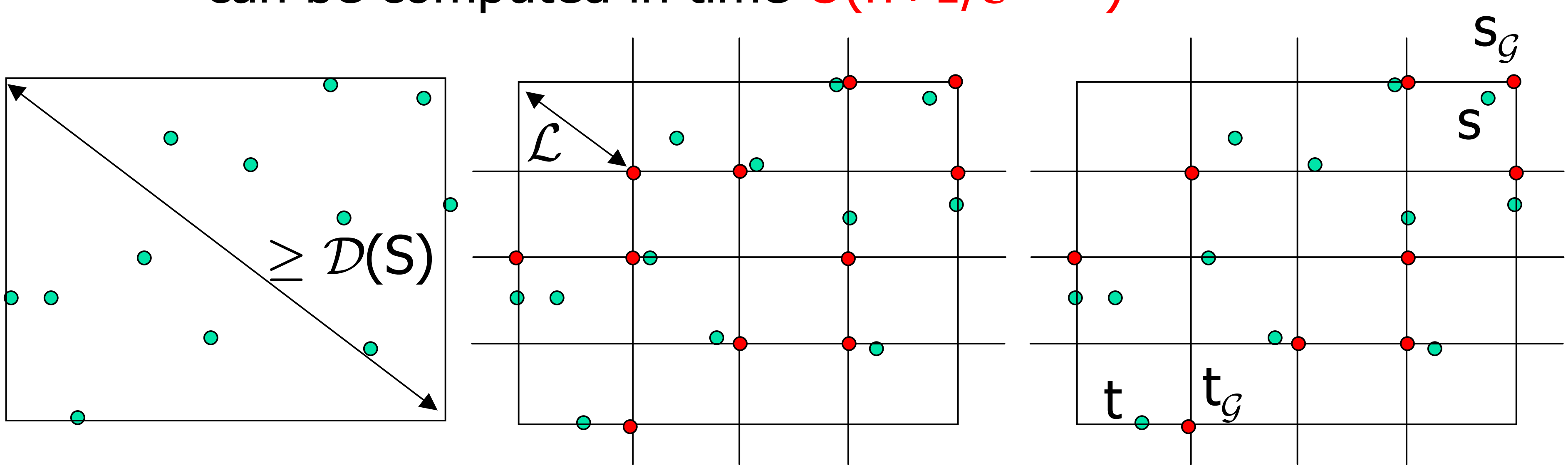
original points S
compute B(S)

snap S to grid($\varepsilon/(2\sqrt{d})$ B)
to get $S_{\mathcal{G}}$

keep only extreme grid points
brute force to find the diameter
($s_{\mathcal{G}}$, $t_{\mathcal{G}}$) and thus (s,t)

(1+ε)-Approximate Algorithm for Diameter

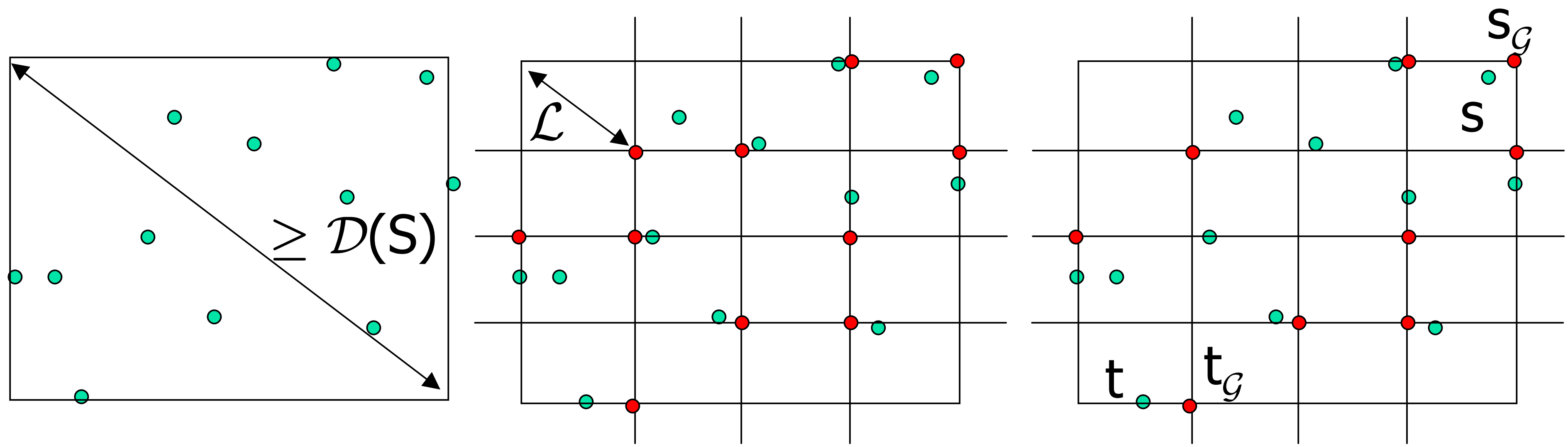
Lemma: A pair of points (s,t) such that $|st| \geq (1-\varepsilon)\mathcal{D}(S)$ can be computed in time $O(n+1/\varepsilon^{2(d-1)})$



original points S compute B(S)	snap S to grid($\varepsilon/(2\sqrt{d})$ B) grid size ~ $O(\varepsilon)$, $\lvert S_{\mathcal{G}}\rvert = O(1/\varepsilon^d)$	keep only extreme grid points ➔ $O(1/\varepsilon^{d-1})$ remains brute force to find the diameter
cost: $O(n)$	cost: $O(n)$	cost: $O(1/\varepsilon^{2(d-1)})$

(1+ε)-Approximate Algorithm for Diameter

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



original points S
compute B(S)

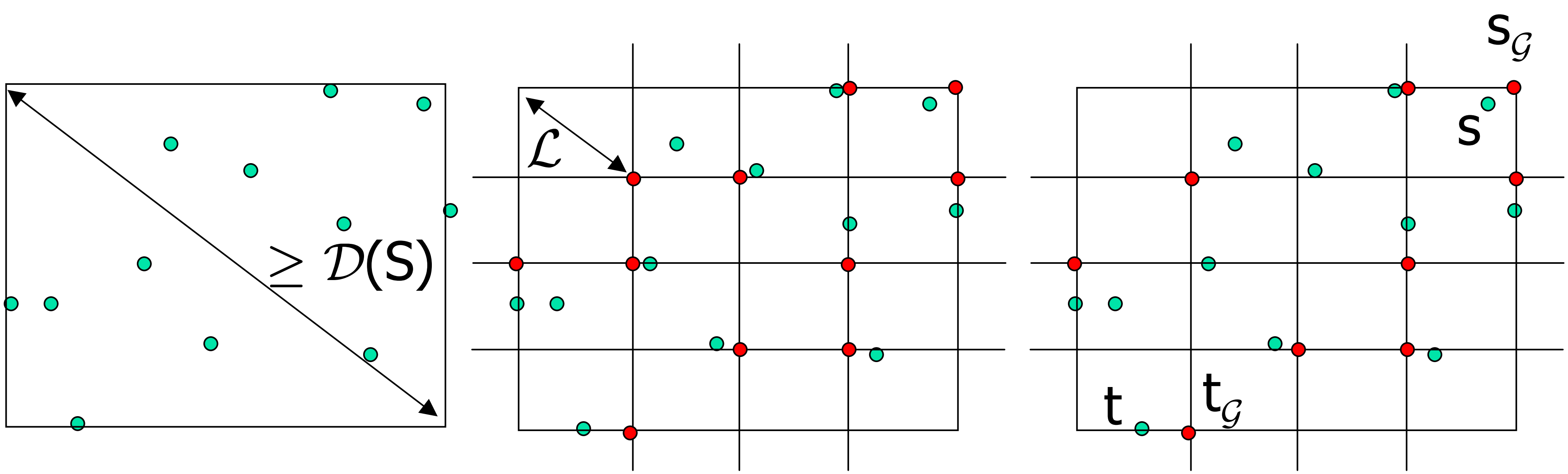
snap S to grid($\varepsilon/(2\sqrt{d})$ B)

keep only extreme grid points
brute force to find the diameter

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

Can we improve?

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



original points S
compute B(S)

snap S to grid($\varepsilon/(2\sqrt{d})$ B)
to get $S_{\mathcal{G}}$

keep only extreme grid points
brute force to find the diameter
$(s_{\mathcal{G}}, t_{\mathcal{G}})$ and thus (s,t)

Computing Extreme Points

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

- $\mathcal{CH}(S_{\mathcal{G}})$ 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:
  $O(h^2) = O(1/\varepsilon^{2(d-1)d/(d+1)})$

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

$O(n + 1/\varepsilon^3)$ in R^3

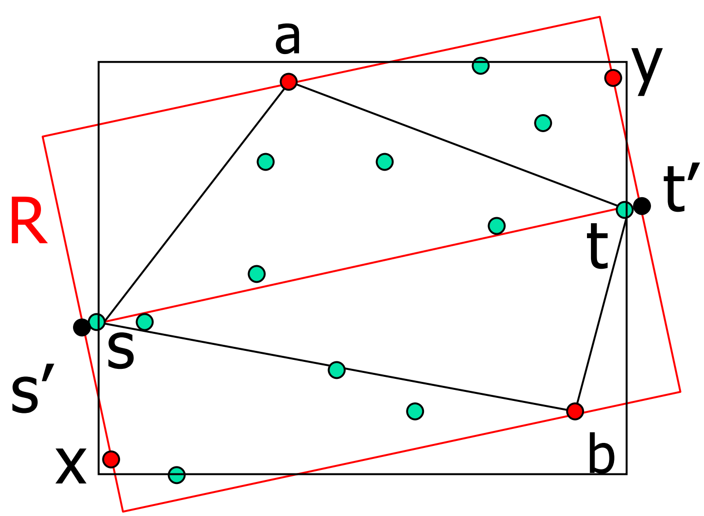
Better Diameter Computation in R^3

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

➔ $O(n + (1/\varepsilon^{3/2})\log(1/\varepsilon))$ time algorithm

Coarse Approximation Algorithm for Bounding Box

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



Idea (in 2D, for point set Q)

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

   the width of R

$\mathrm{Area}(R(Q)) = 2\,\mathrm{Area}(s'bt'a) \leq 2\sqrt{2}\,\mathrm{Area}(sbta) \leq 2\sqrt{2}\,\mathcal{CH}(Q) \leq 2\sqrt{2}\,\mathrm{Area}(R_{opt}(Q))$

$\mathrm{Vol}(B(S)) \leq 6\sqrt{6}\,\mathrm{Vol}(\mathcal{CH}(S))$

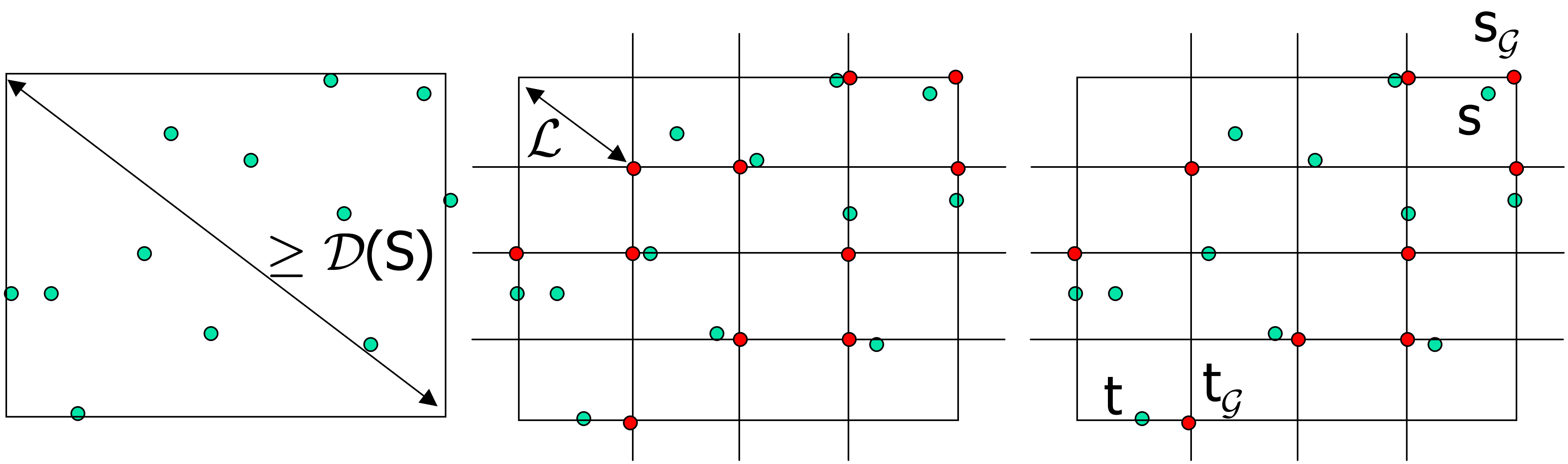
Coarse Box Property

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

- Proof idea:
  - Suppose B(S) is an axis-aligned unit cube
  - $\mathrm{Vol}(\mathcal{CH}(S)) \geq 1/(6\sqrt{6})$
  - 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 $|st| \geq (1-\varepsilon)\mathcal{D}(S)$ can be computed in time $O(n+1/\varepsilon^{2(d-1)})$



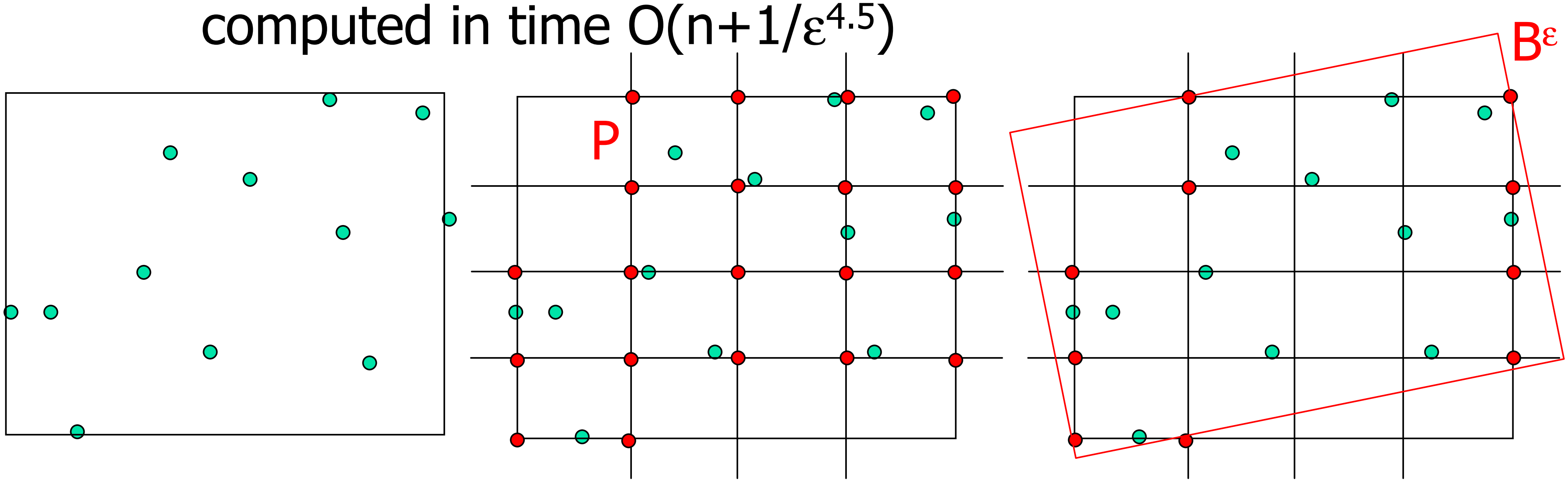
original points S
compute B(S)

snap S to grid($\varepsilon/(2\sqrt{d})$ B)
to get $S_{\mathcal{G}}$

keep only extreme grid points
brute force to find the diameter
($s_{\mathcal{G}}$, $t_{\mathcal{G}}$) 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

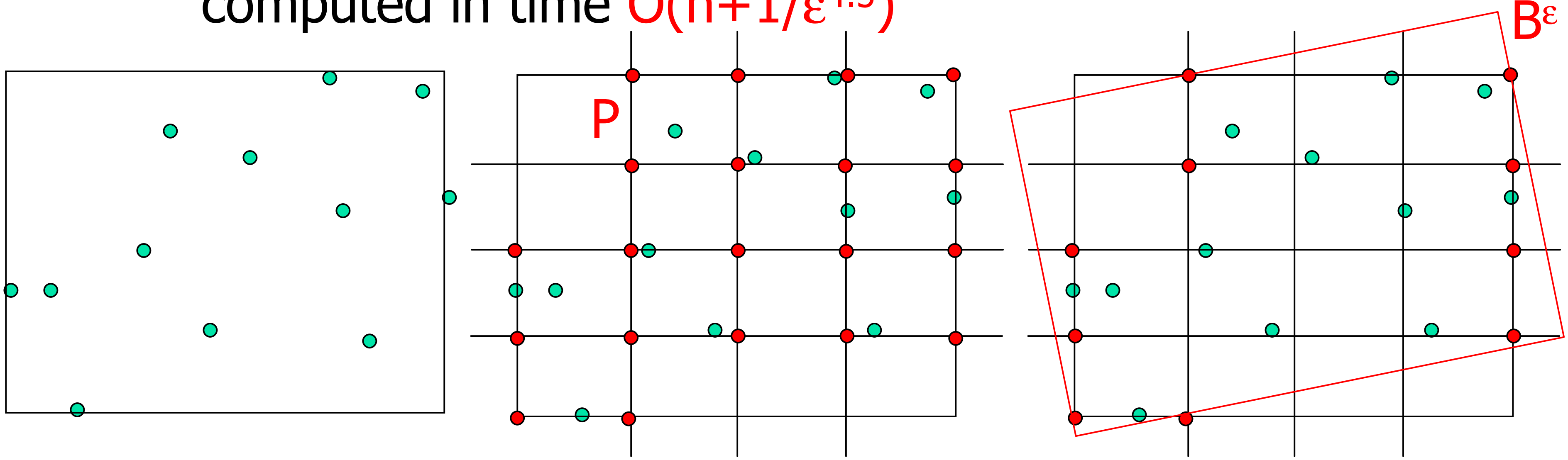
expand S to grid($\varepsilon/428$ B)
to get $S_{\mathcal{G}}$

keep only extreme grid points
and their convex hull

brute force to find the
bounding box B^ε_{opt}

$(1+\varepsilon)$-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

Cost: $O(n)$

expand S to grid($\varepsilon/428$ B)
to get $S_{\mathcal{G}}$

Cost: $O(n)$

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

Compute: B^{ε}_{opt}
Cost: $O(1/(\varepsilon^{3/2})^3)$

$(1+\varepsilon)$-Approximate Algorithm for Bounding Box

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

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

We don't know $B_{opt}(S)$, yet we know B^ε_{opt} exists!

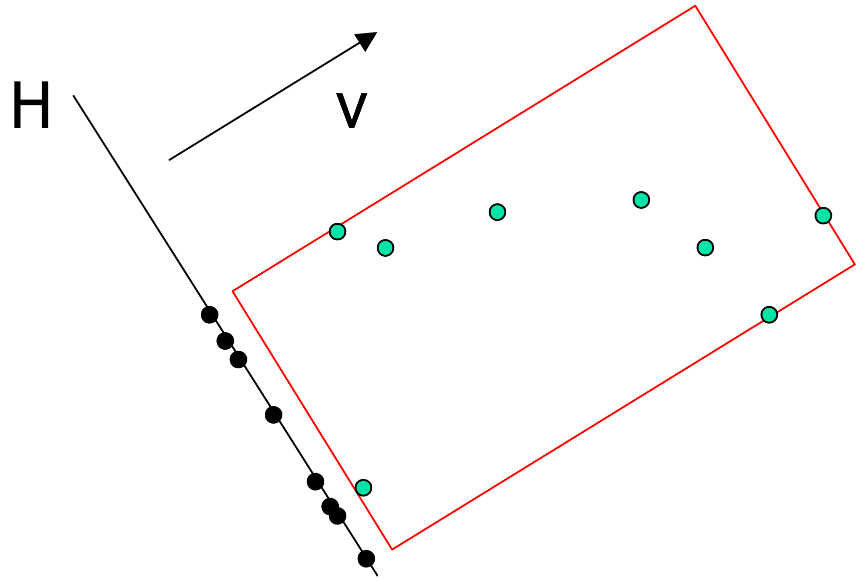
- $$\begin{aligned} P &\subset \mathcal{CH}(S) \oplus B^\varepsilon \\ &\subset \mathcal{CH}(S) \oplus B^\varepsilon_{opt} \\ &\subset B_{opt}(S) \oplus B^\varepsilon_{opt} \\ &= (1+\varepsilon/4)\, B_{opt}(S) \end{aligned}$$
- $Vol(B_{opt}(P)) \leq (1+\varepsilon/4)^3\, Vol(B_{opt}(S)) \leq (1+\varepsilon)\, B_{opt}(S)$

Algorithm is too complicated to implement!

Grid Search Algorithm

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

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



The Algorithm

ALGORITHM GRIDSEARCHMINVOLBBX (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 $B_{\mathrm{opt}}(S)$.

begin

    Compute $\mathcal{CH}(S)$; ← O(n log n)
    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 min_vol $:= \infty$ and $v^* :=$ undefined;
    for $v \in$ BG **do**
        Compute $B = B_{\mathrm{opt}}(S, \{v\})$; ← O(n)
        if min_vol $> \mathrm{Vol}(B)$ **then do**
            Set min_vol $:= \mathrm{Vol}(B)$ and $v^* := v$;
        od
    end for
    Return $B_{\mathrm{opt}}(S, \{v^*\})$;

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_{opt}

- The optimal box with direction v containing B_{opt} has volume only slightly larger than B_{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

along appx diagonal

min-vol along edges

grid search, along B*

search size

reprojection

grid search, std. coord.

$\|S\|$	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

Results – 48 points

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

Results – 100 points

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

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