入力誤差を考慮した幾何アルゴリズム

Geometric Algorithms with Imprecise Input

トポロジーとコンピュータ2005
11月30日

鳥取環境大学　情報システム学科
助教授　永井孝幸

What is Computational Geometry?

- 計算幾何学(けいさん・きかがく≠けいさんき・かがく)
- A field of computer science
  - analyse geometric problems
  - design/implement *efficient* algorithms
- Application areas are:
  - computer graphics, computer aided design, circuit layout, robotics, GIS,etc.

Today's topics

- Part1　Introduction
  - Basic ideas in Computational Geometry
  - Convex hull and related problems
- Part2　Algorithms with imprecise input
  - Difficulties with imprecise input
  - Revisit convex hull and related problems

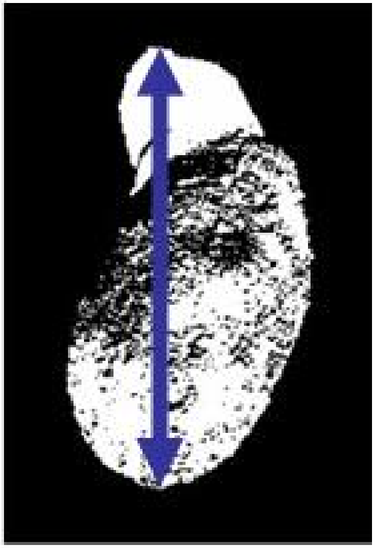
All discussions are in 2-d Euclidean space.

Problem: diameter of points

- Find the maximum distance R between points in S
  - Computation itself is trivial.

$$R = \max_{p,q \in S} \{ d(p,q) \}$$

  - But, *efficient* computation is NOT trivial.



イトカワ画像**(by JAXA)**

What is the *performance* of algorithm ?

- number of primitive operations
  - Add/sub/mul/div, comparison, conditional branch
  - Given by a function of input size n

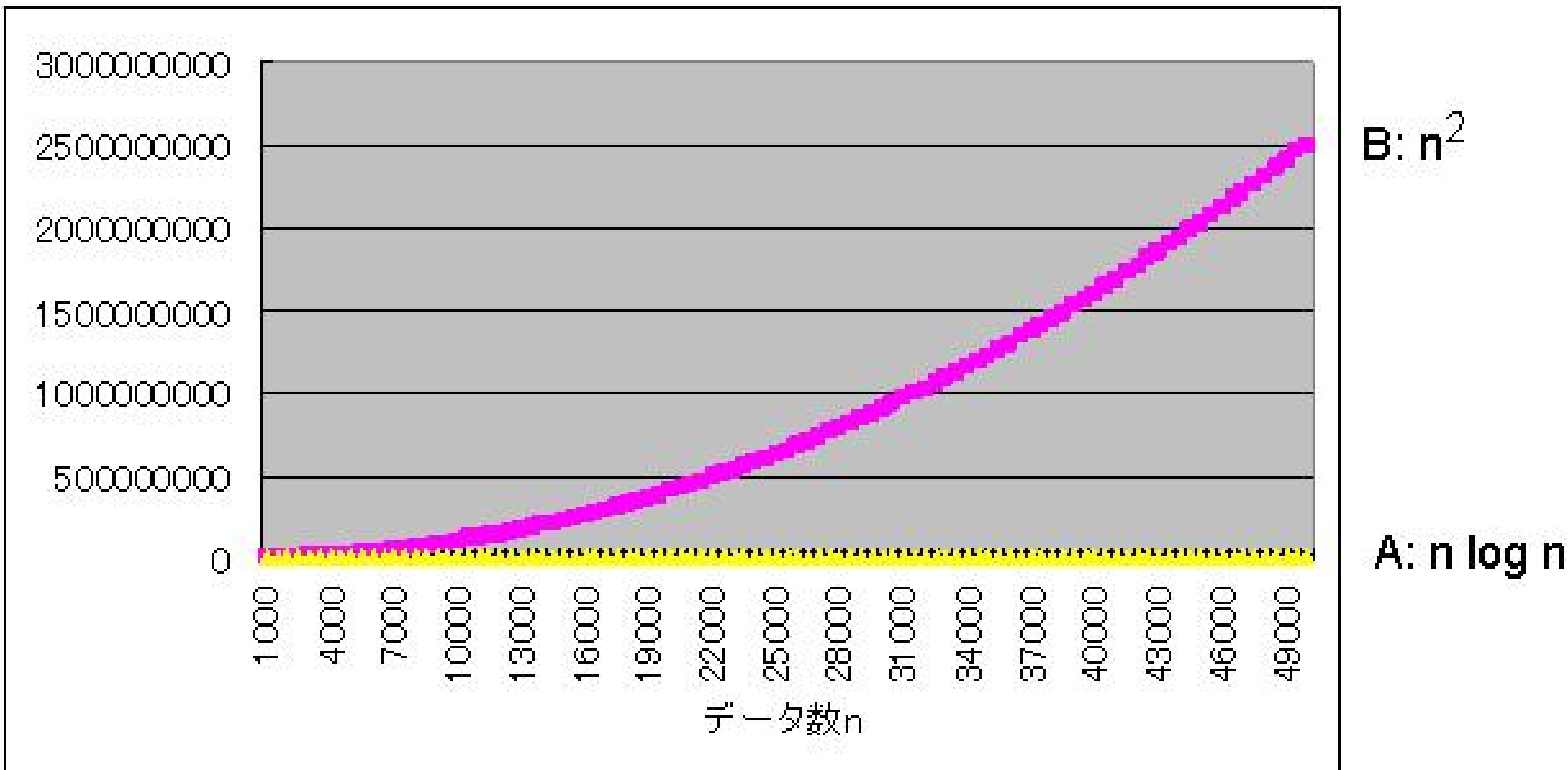
  ex. Bubble-sort sorts n numbers in $O(n^2)$ time.

  ex. Merge-sort sorts n numbers in $O(n \log n)$ time.

- Basic 2-d problems in Computational Geometry have $O(n \log n)$-time solutions.

The impact of efficient algorithm

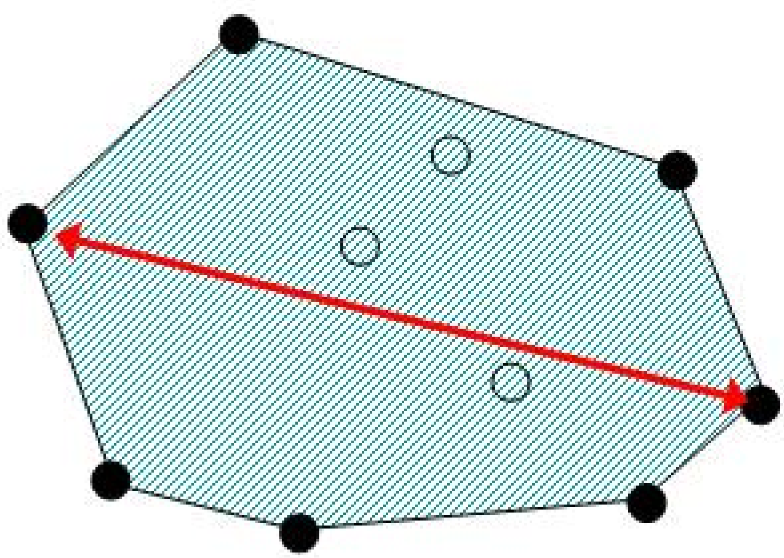
- A: n log n vs B:n^2
  - for $n=10^5$, A is 6,020 times faster than B.

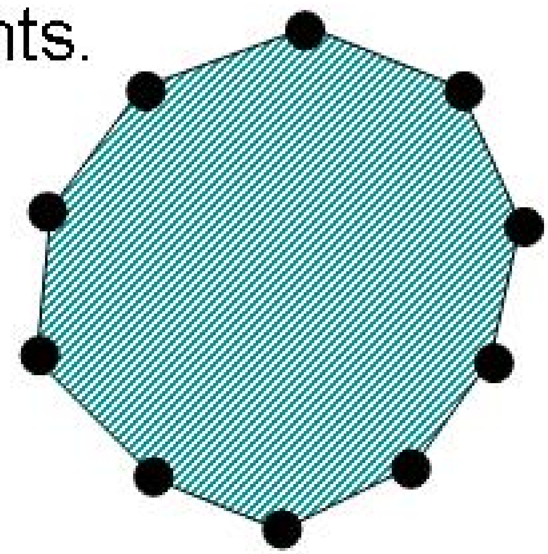


What determines diameter R?

- *R is determined by vertices of convex hull*
  - Convex hull *CH(S)*:

    The smallest convex polygon that contains all points of S
  - We can discard interior points.

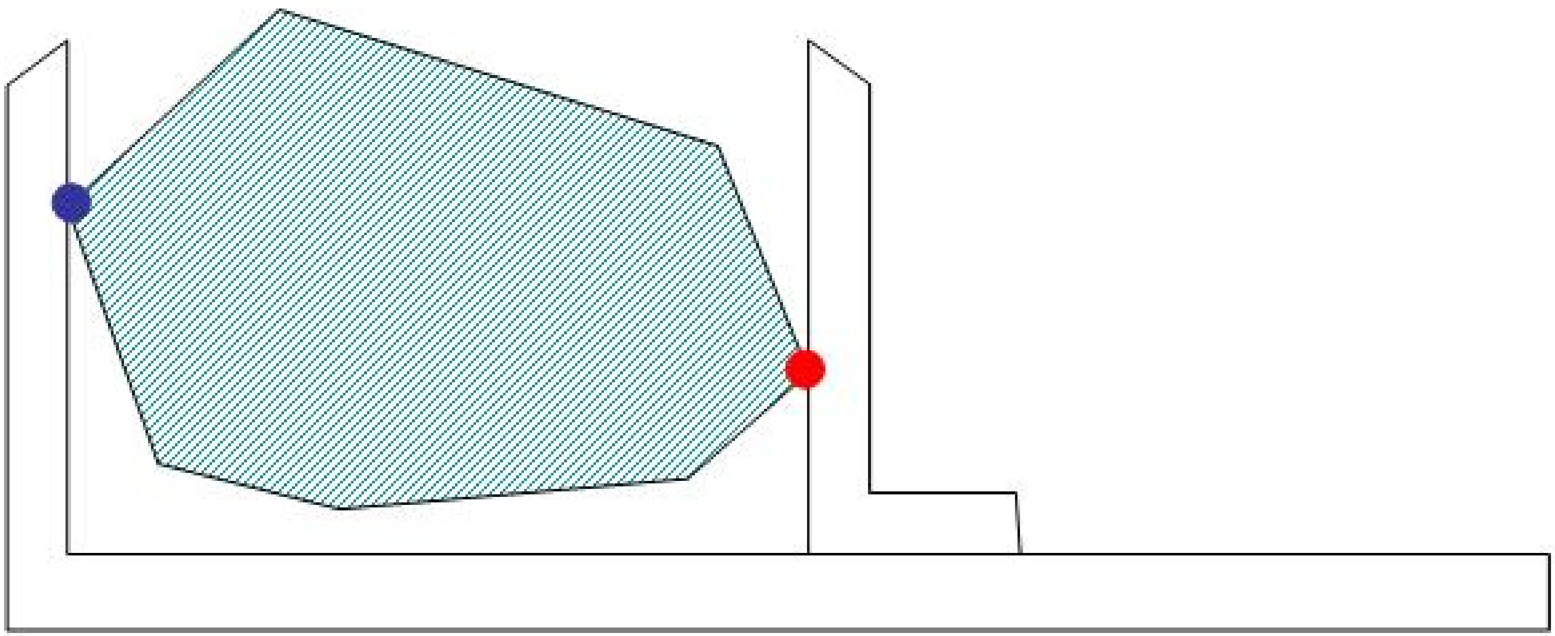


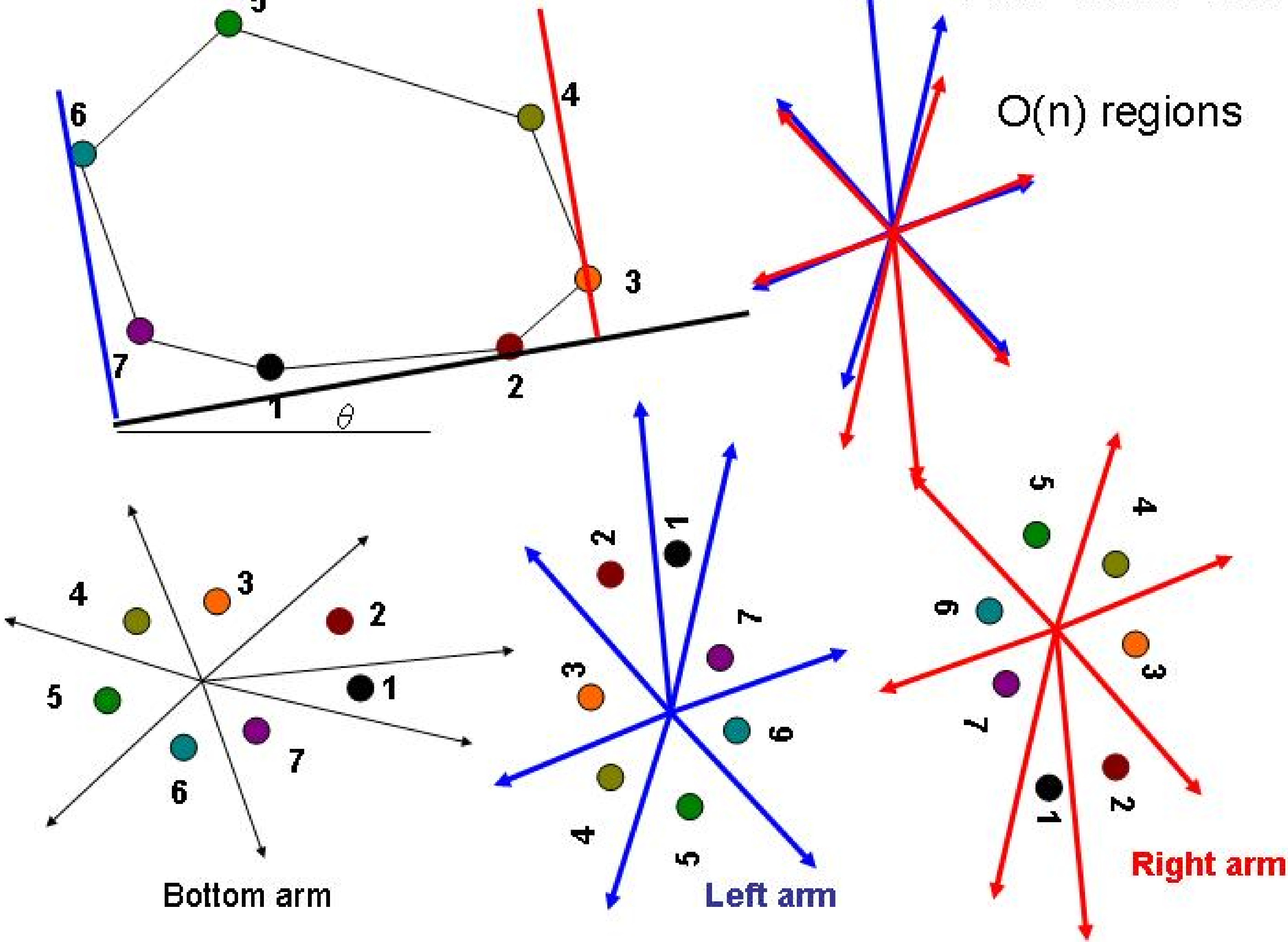


How about this case ?

Rotating caliper method

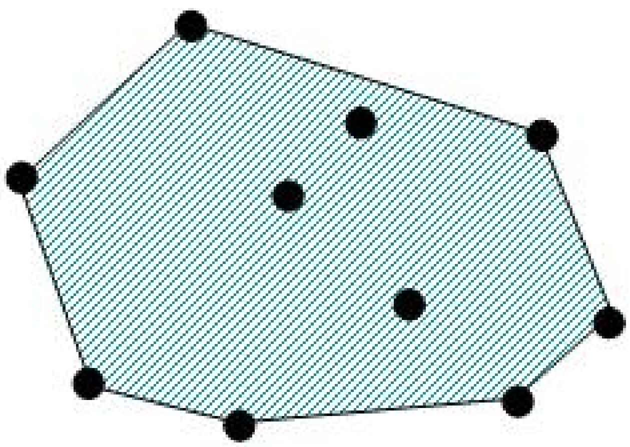
- Diameter = the maximum of width
  - Width is given by antipodal pairs.
  - Q. How many pairs are there ? A. O(n)

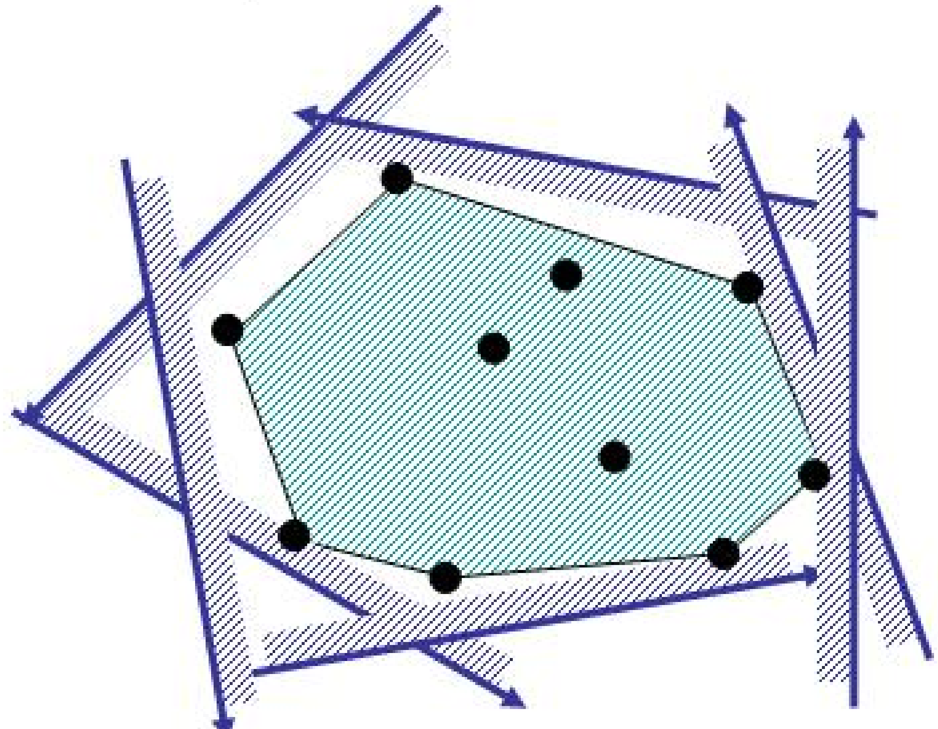



O(n) regions
θ
Bottom arm
Left arm
Right arm

Computation of convex hull

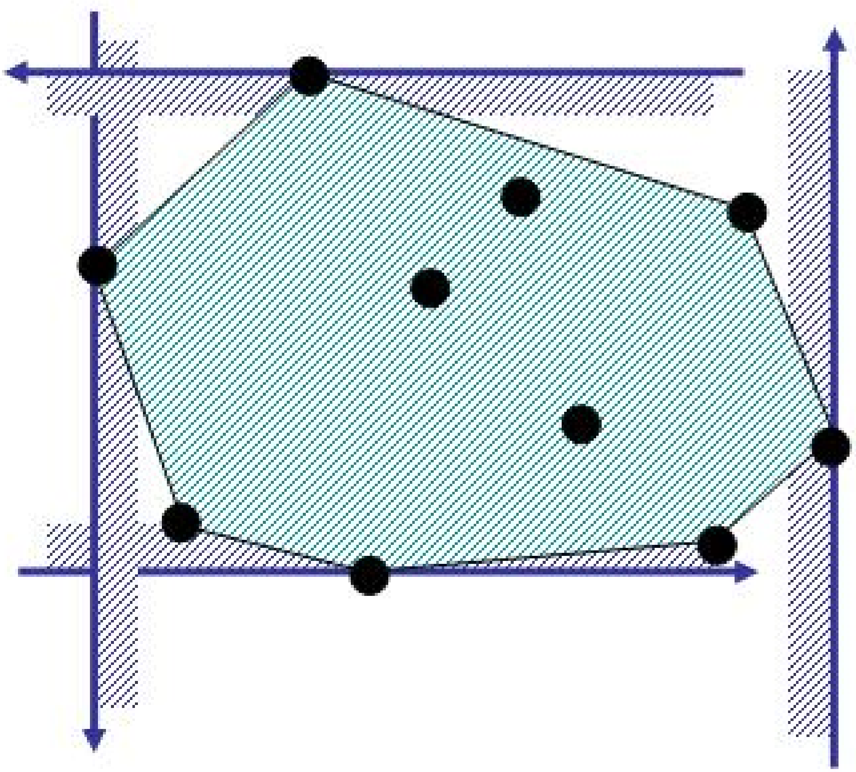
- How do we characterize convex hull of S?
  1. The smallest convex polygon that encloses S
  2. The intersection of all convex sets enclosing S
  3. The intersection of all half-planes that contain S





Observation 1

- If a half-plane that touches *p* contains all points of *S*, *p* is a vertex of convex hull.
  - Ex. Leftmost point of S is a vertex of CH(S).

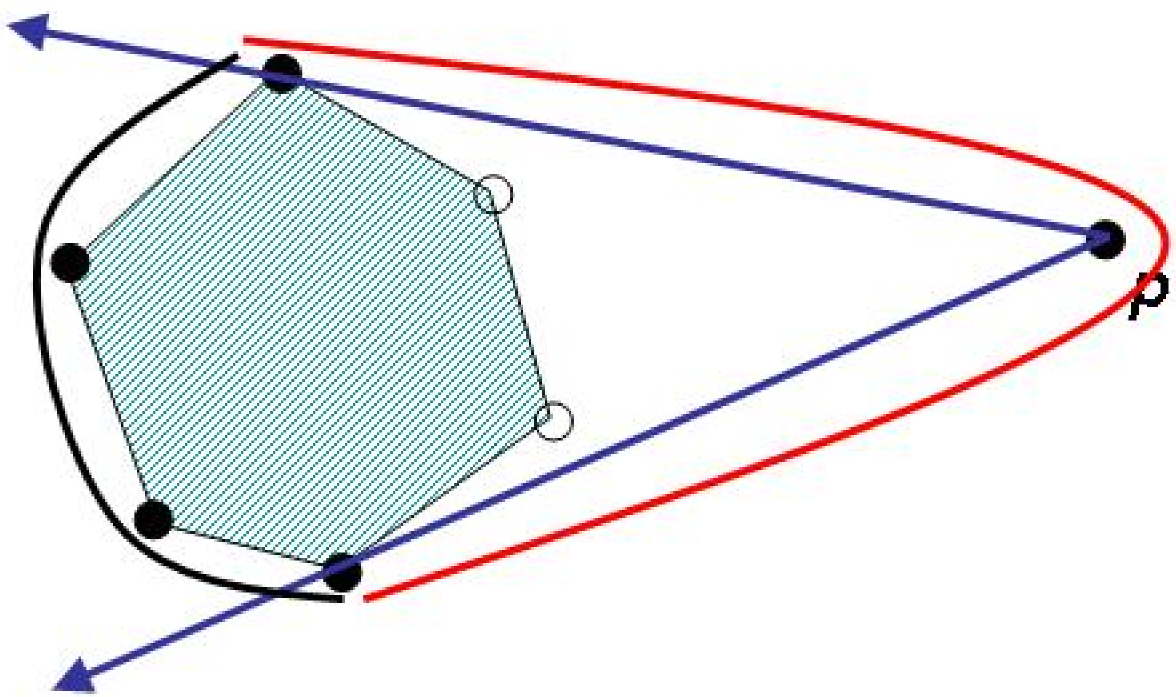


Observation 2

- Let p be a point outside of CH(S). Then,

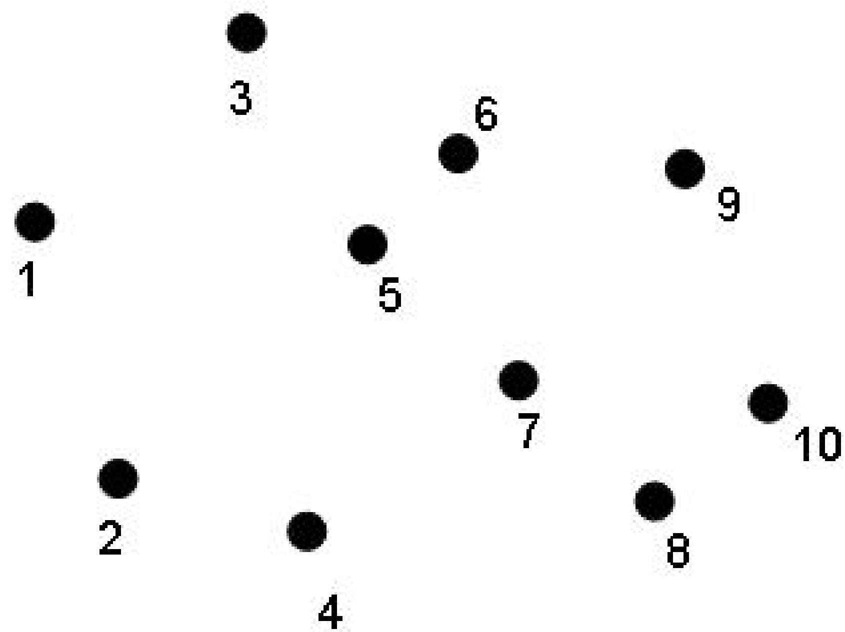
  (1) p is a vertex of CH(S ∪ {p}) , and

  (2) we can find vertices of CH(S ∪ {p}) by drawing support lines from p to CH(S).



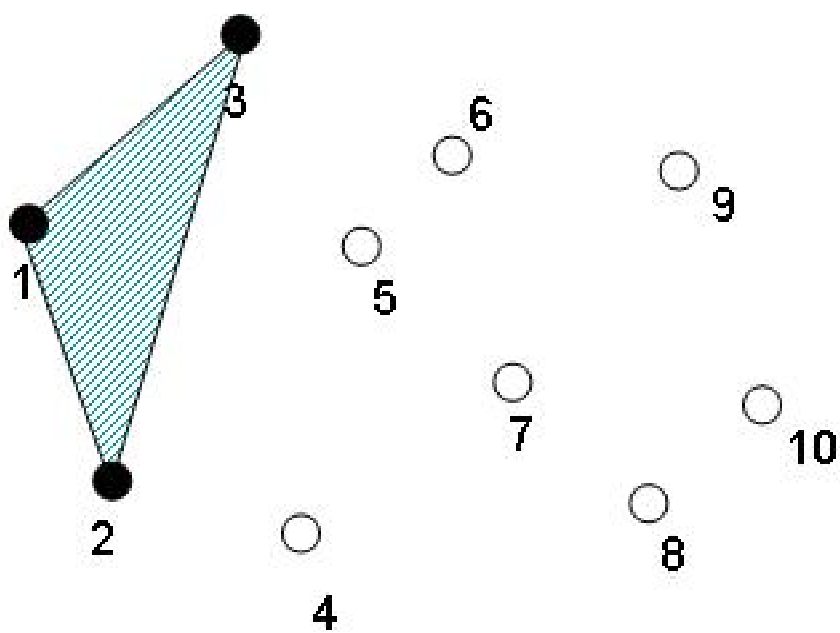
Incremental construction(1)

1. Sort points according to x-coordinates
   - $S=\{p_1, p_2, p_3, \ldots, p_n\}$, $P_i=\{p_1, p_2, \ldots, p_i\}$



Incremental construction(2)

2.Compute CH(P_3) (*i.e. triangle*)

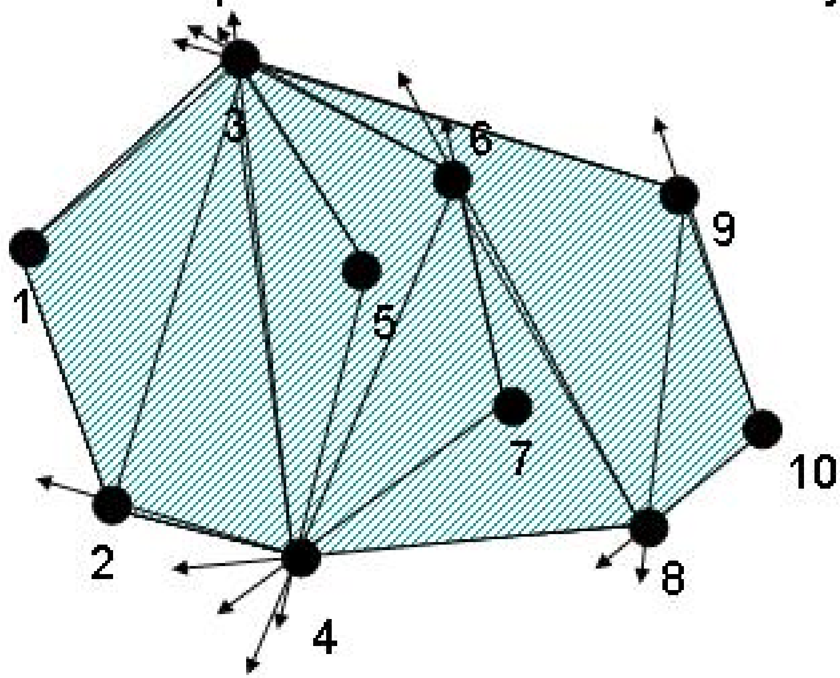


Incremental construction(3)

3.Compute $CH(P_{i+1})$ i.e. $CH(P_i \cup \{p_{i+1}\})$

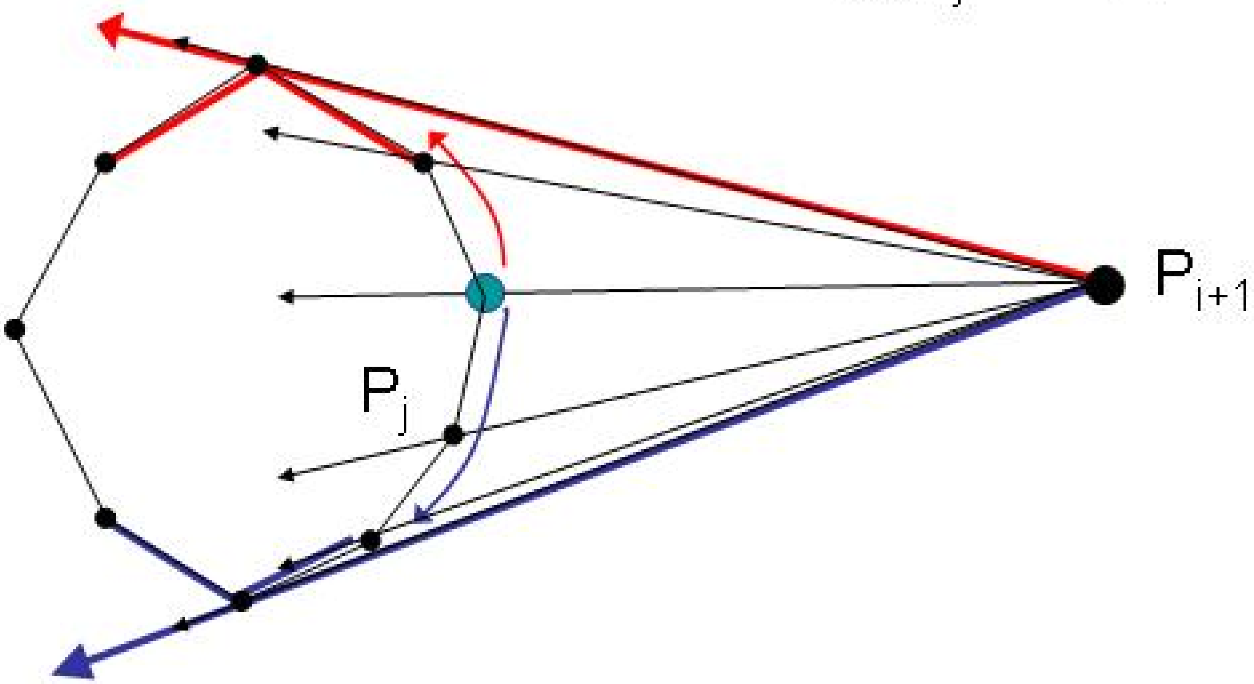
– 3-1.Find support lines from pi+1 to CH(Pi)

– 3-2.Update the boundary of convex hull



Computation of support line

- Traverse the boundary from the rightmost vertex
  - We can determine if line $p_{i+1}p_j$ is support in O(1) time.



Total computational time

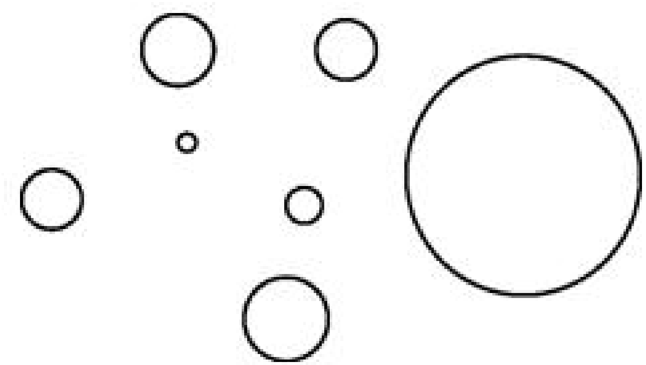
1. Convex hull of n points in plane
   - Sort n points — O(n log n)
   - Compute initial triangle — O(1)
   - Update boundary n-3 times — O(n)

   Total O(n log n) time
2. Diameter of n point in plane
   - compute convex hull — O(n log n)
   - check all antipodal pairs — O(n)

   Total O(n log n) time

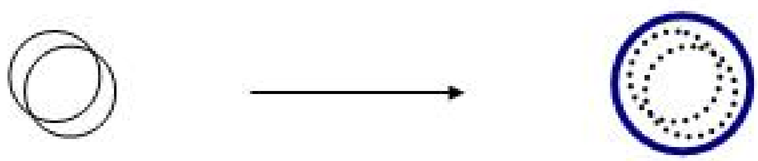
Part2 Algorithms with imprecise input

- Location of points are known only up to a limited accuracy.
  - Measurement / round-off / approximation error
- Solution
  - Guarantee the accuracy of computed results

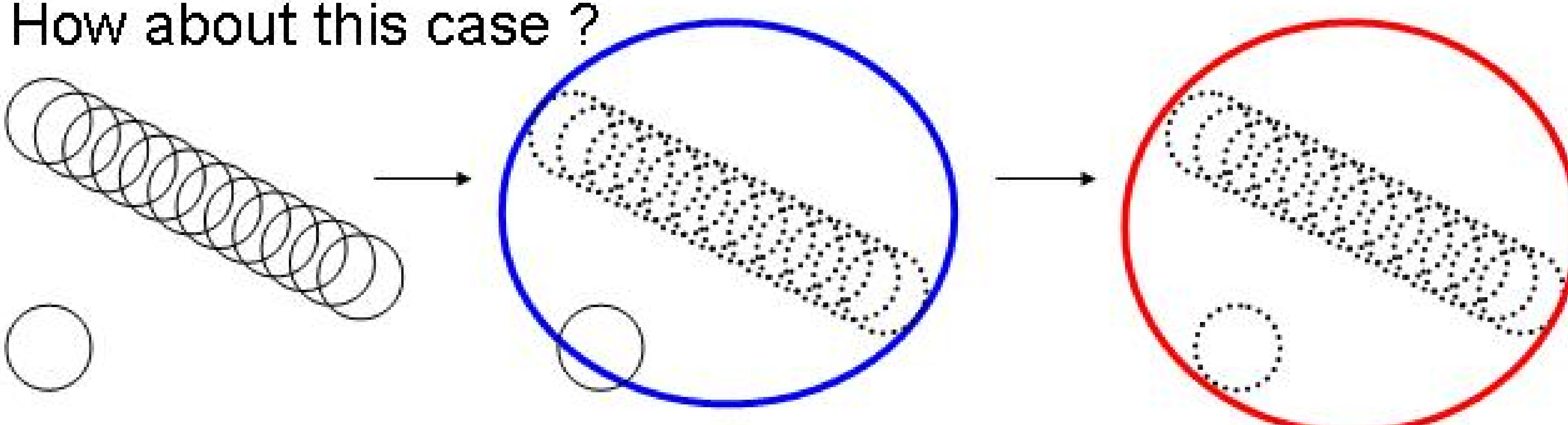


Ad hoc solution does not work

- "If two points are very close, we merge them."
  - Is it OK?



How about this case ?

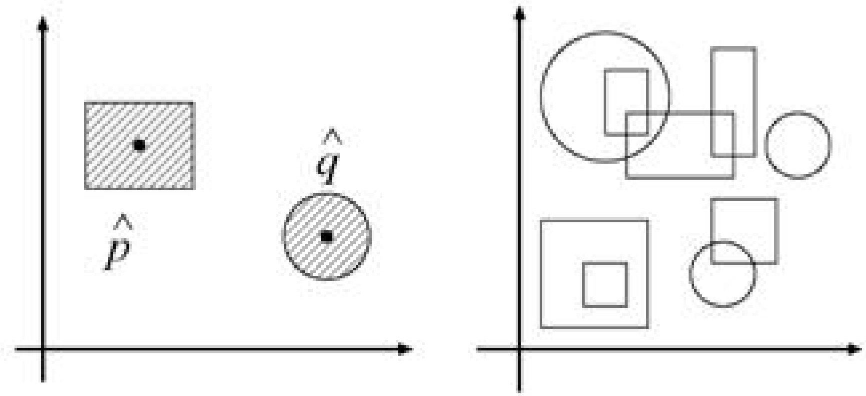


In general, merged point can be arbitrary large…

Input model

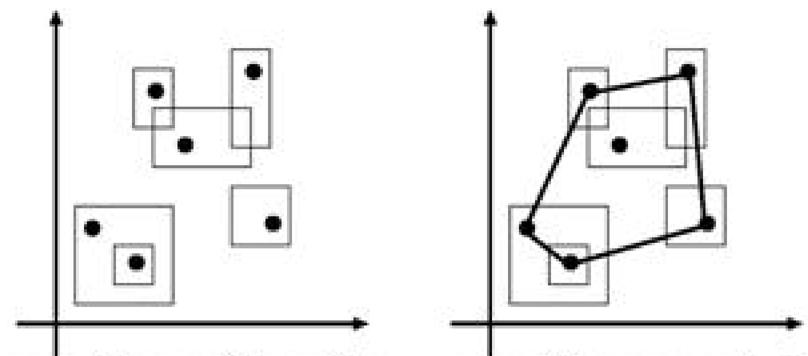
- ε -point:

“point given by a convex region”



- Possible object X

“an object realized in a possible configuration of points”

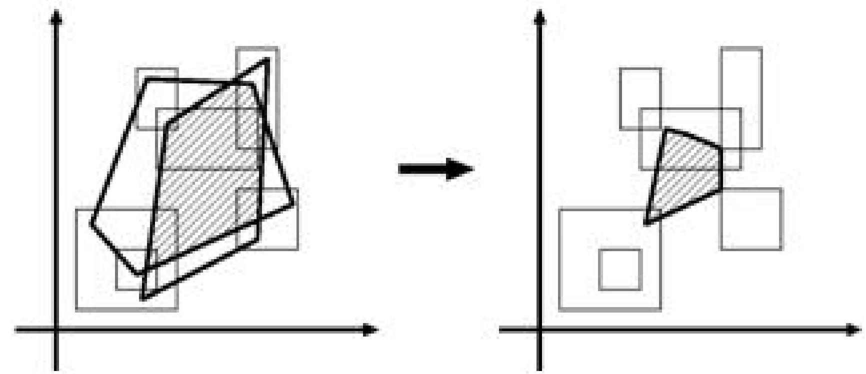


possible configuration ***possible convex hull***

Error bound of convex hull

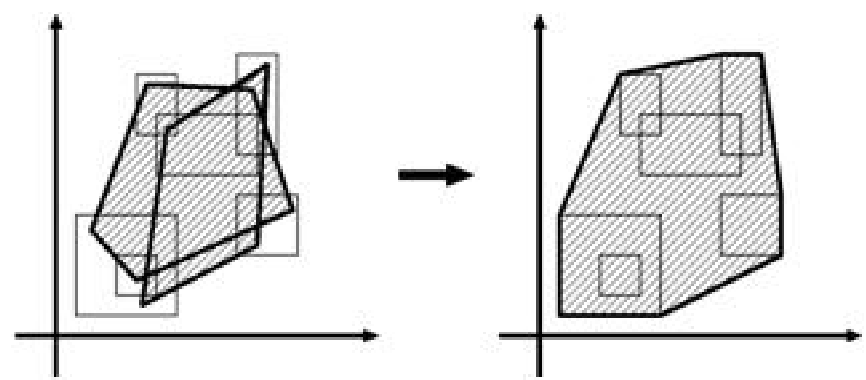
- Inner convex hull $\underline{conv}(S)$

  $= \cap$ {all possible hulls for S}



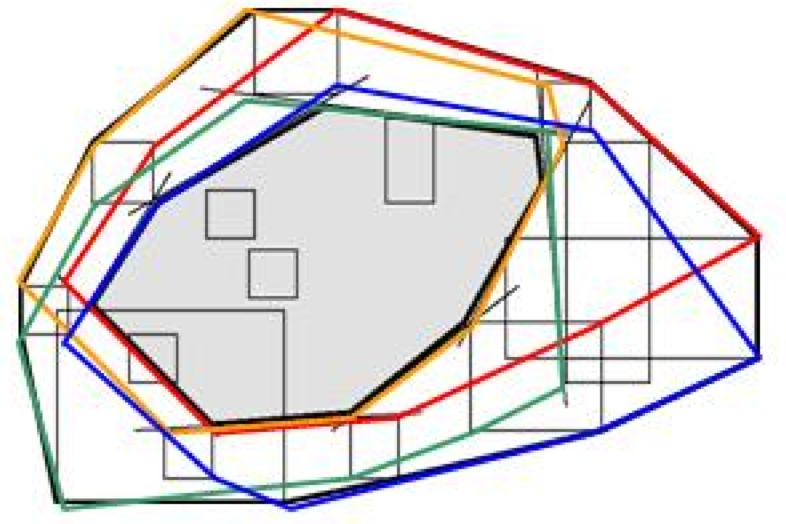
- Outer convex hull $\overline{conv}(S)$

  $= \cup$ {all possible hulls for S}



Case: rectangle ε -Point

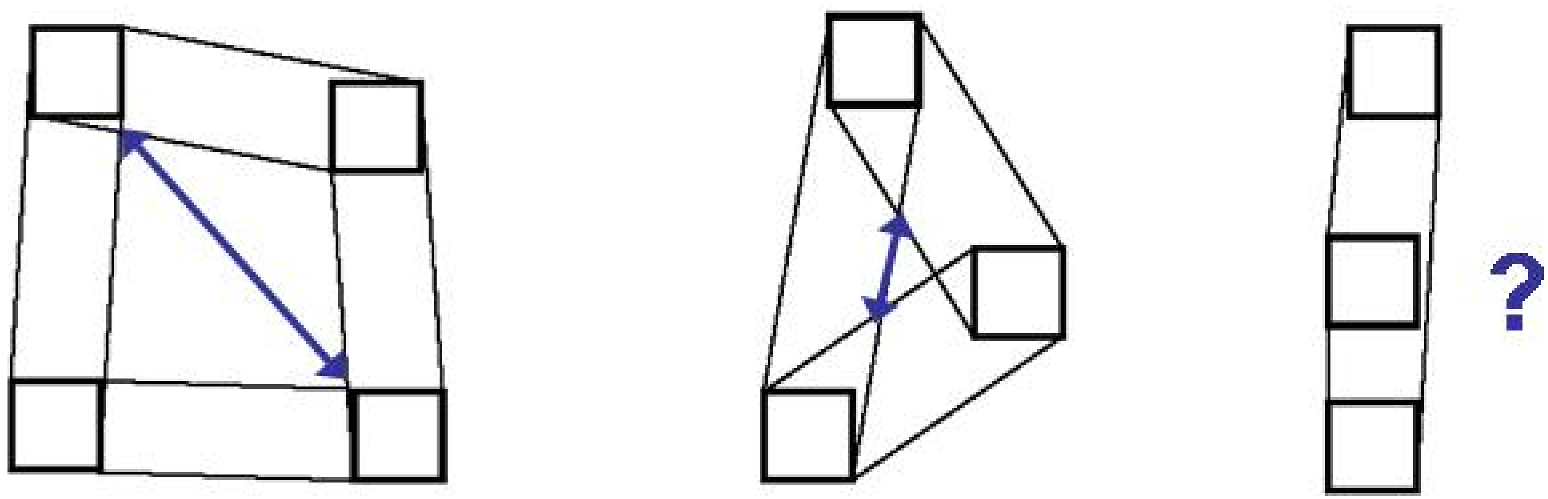
- Outer convex hull
  - Convex hull of ε -Points
- Inner convex hull
  - Intersection of 4 convex hulls



How about other input model ?
How about other problem ?

Error bound of diameter

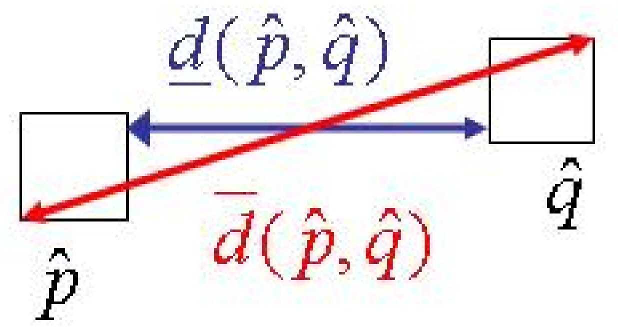
- The boundary of inner convex hull is NOT useful



Tight error bound is not necessarily obtained

O(n^2) algorithm again ?

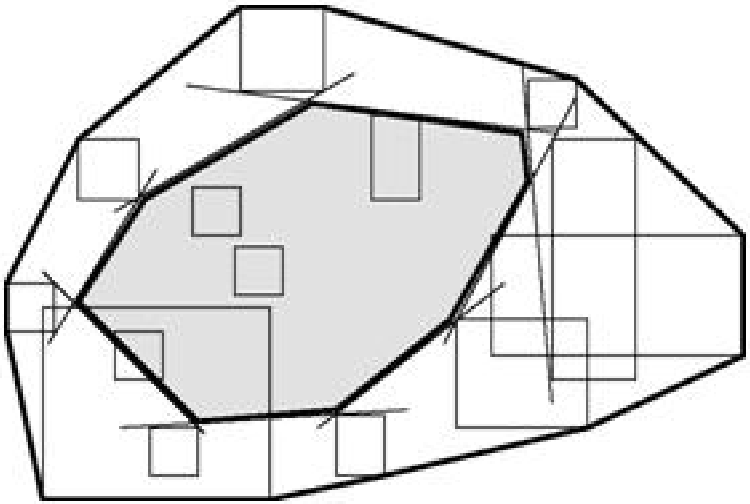
Compute possible distances between ε -points,
then possible diameters are given as follows:



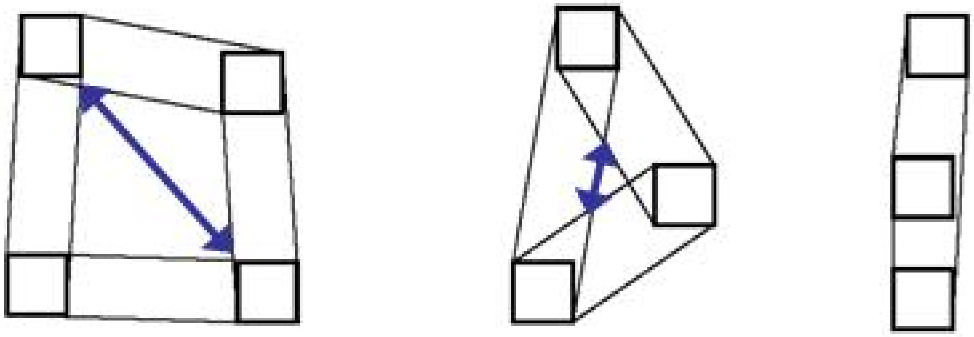
$$\underline{R} = \max_{\hat{p},\hat{q}\in S}\{\underline{d}(\hat{p},\hat{q})\} \qquad \overline{R} = \max_{\hat{p},\hat{q}\in S}\{\overline{d}(\hat{p},\hat{q})\}$$

Our Idea

- What's wrong ?
  - The idea of inner/outer convex hull seems useful
  - Boundary representation of them will lose desired information

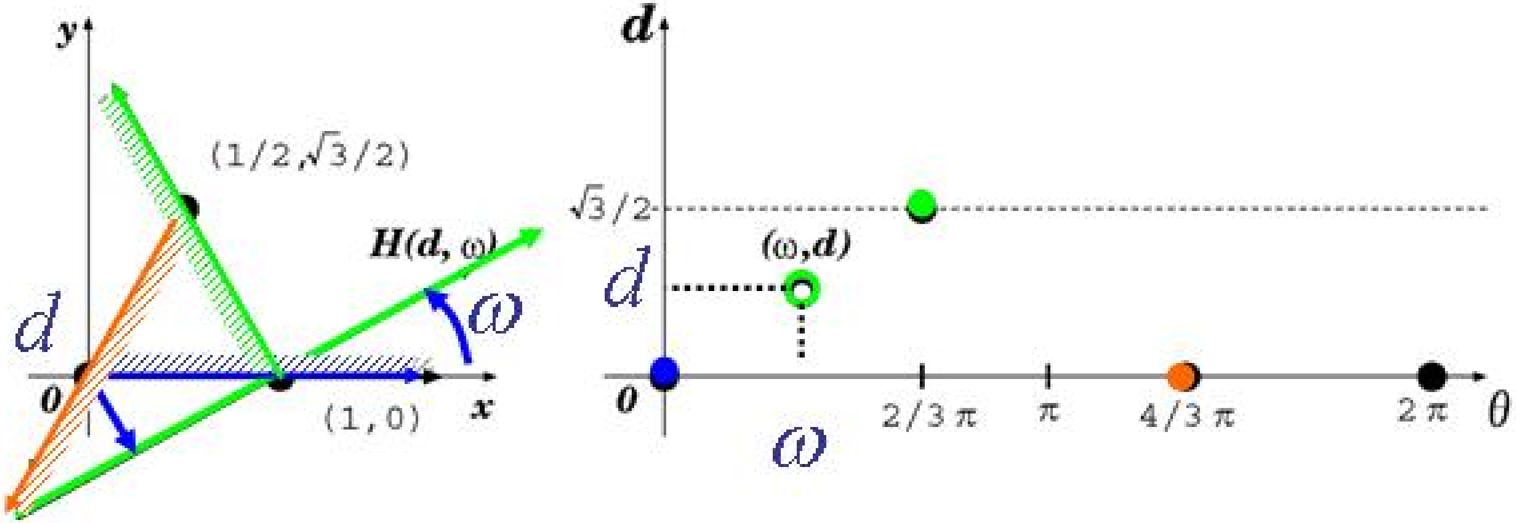


- New idea
  - Characterize a convex region by half-planes.



Half-planes give a convex region

- Convex region = intersection of halfplanes



•Half-plane representation(HPR) of region R

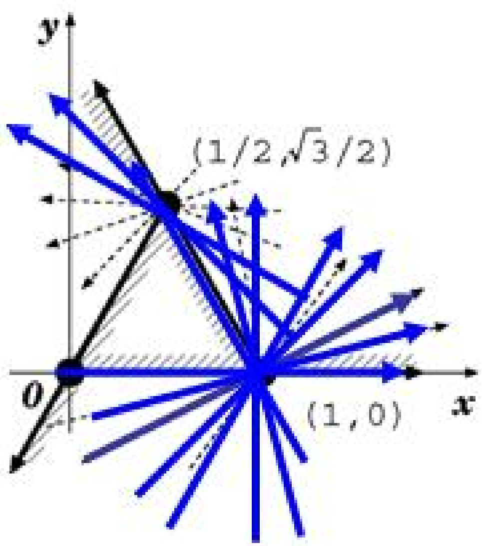
Function $f(\theta)$ is an HPR of R if

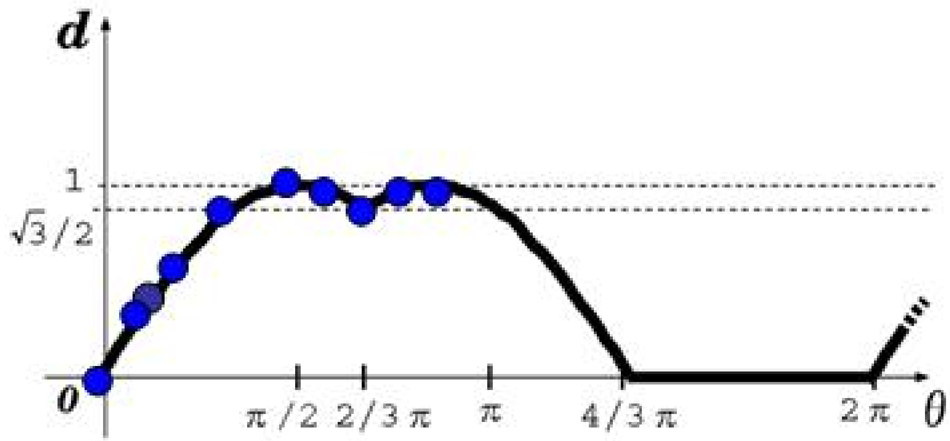
$$\bigcap_{0 \le \theta < 2\pi} H(f, \theta) = R$$

Normalized HPR

- Normalized HPR $d(R,\theta)$ of R

$f(\theta)$ is *normalized* if $H(f,\theta)$ supports R for $\forall\theta$

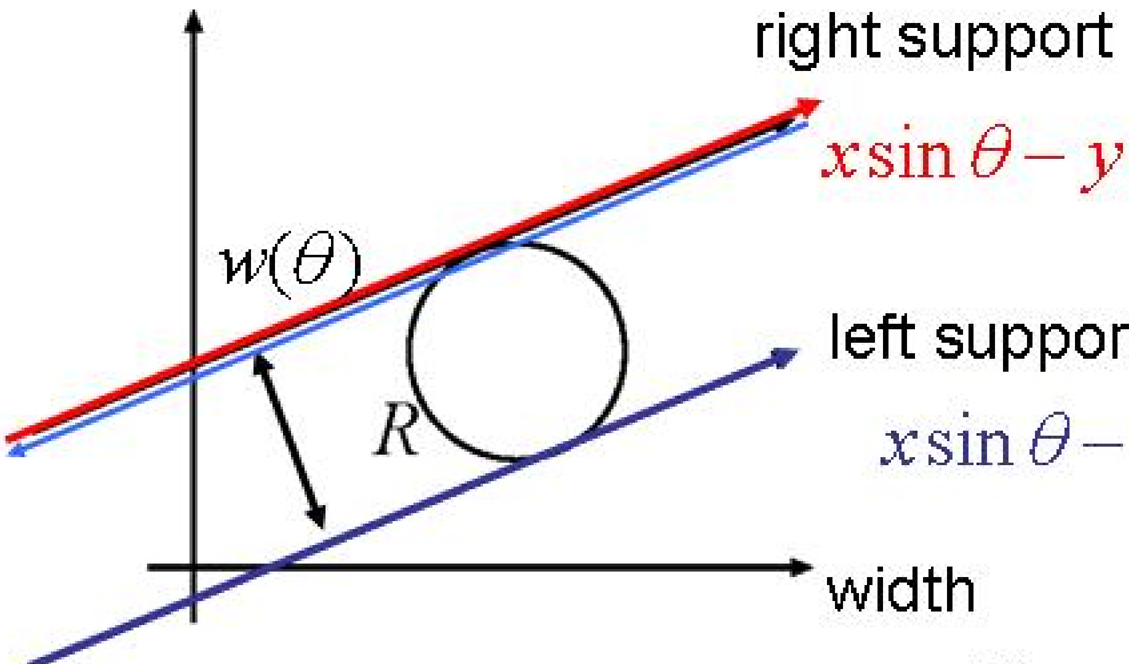




Normalized HPR contains many information.

Properties of HPR

- Normlized HPR gives left/right supports

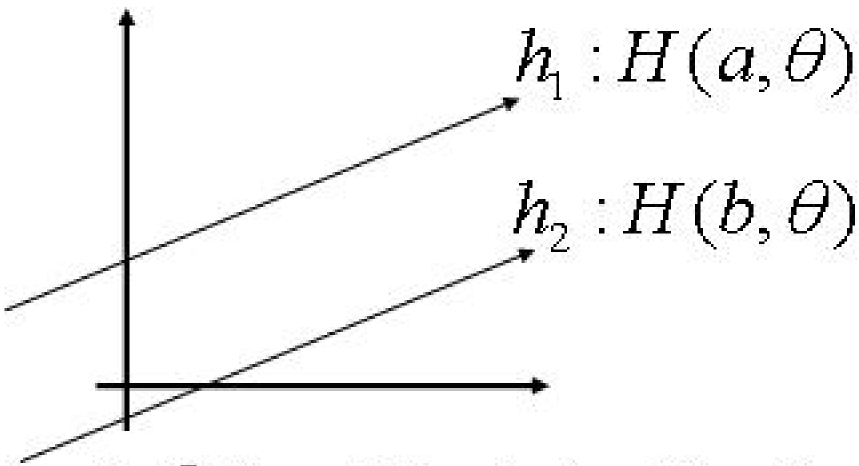


$$x\sin\theta - y\cos\theta = -d(R,\theta+\pi)$$

$$x\sin\theta - y\cos\theta = d(R,\theta)$$

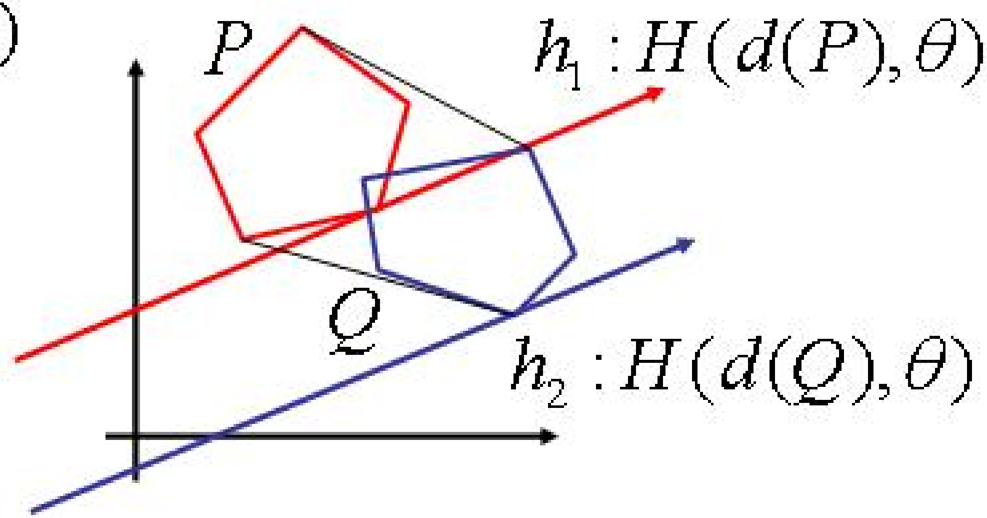
$$w(\theta) = d(R,\theta) + d(R,\theta+\pi)$$

Operations on HPR



$$h_1 \cap h_2 : H(\min\{a, b\}, \theta)$$

$$h_1 \cup h_2 : H(\max\{a, b\}, \theta)$$



$$P \cap Q : \min\{d(P, \theta), d(Q, \theta)\}$$

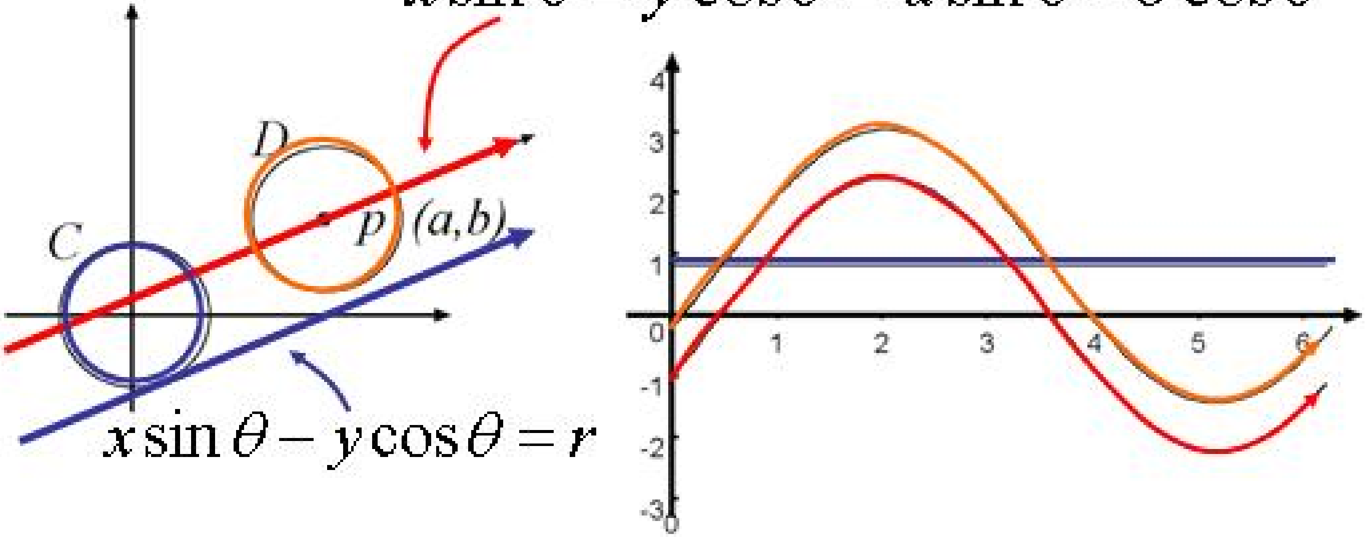
$$conv(P \cup Q) : \max\{d(P, \theta), d(Q, \theta)\}$$

$$P \oplus Q : d(P, \theta) + d(Q, \theta)$$

Examples of HPR

- Point/circle

$$x\sin\theta - y\cos\theta = a\sin\theta - b\cos\theta$$



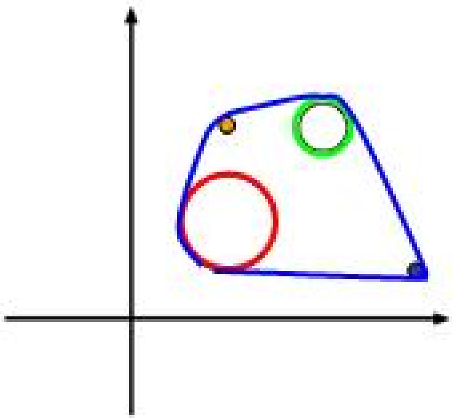
$$d(C,\theta) = r$$

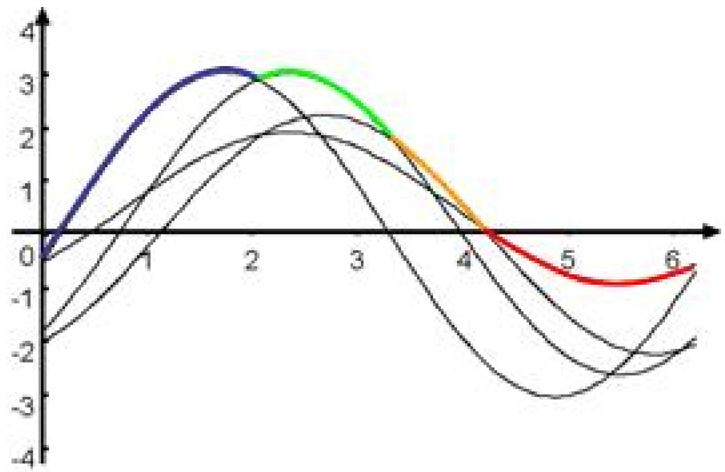
$$d(p,\theta) = a\sin\theta - b\cos\theta$$

$$d(C \oplus p,\theta) = a\sin\theta - b\cos\theta + r$$

Examples of HPR

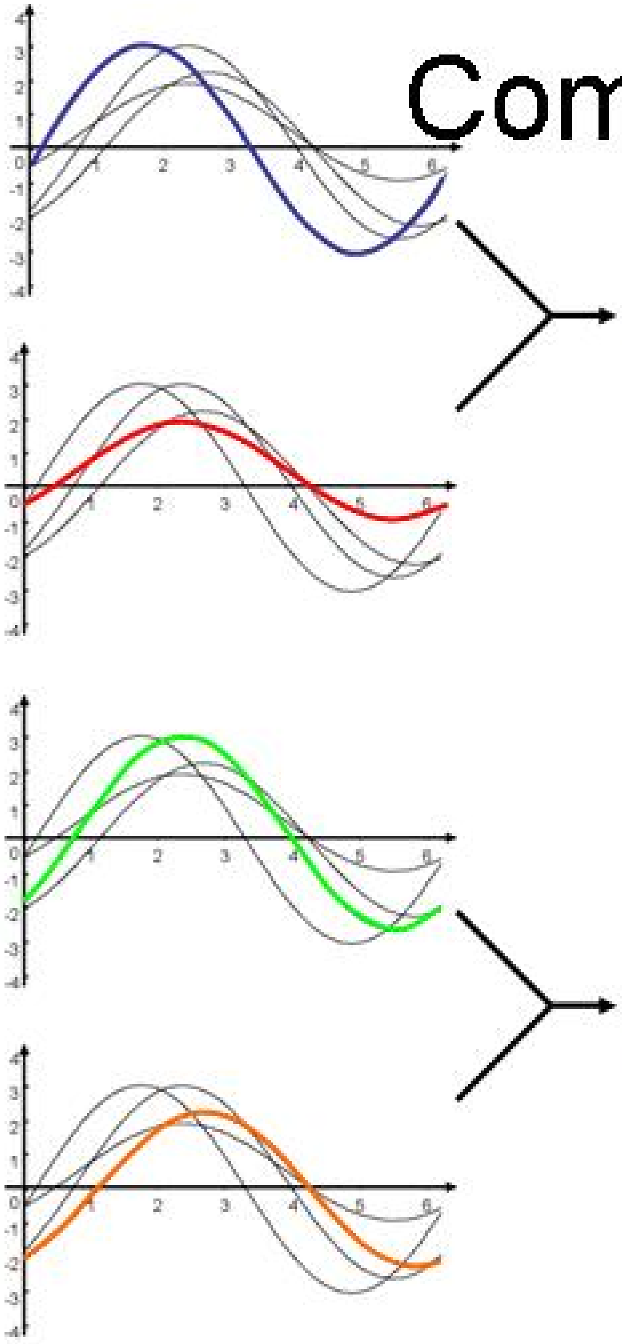
- Convex hull of points and circles

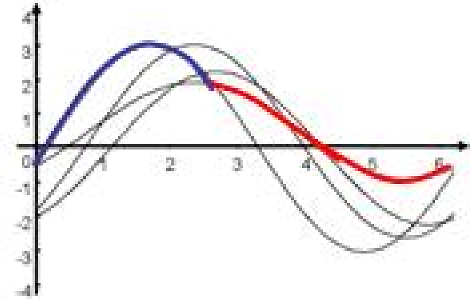




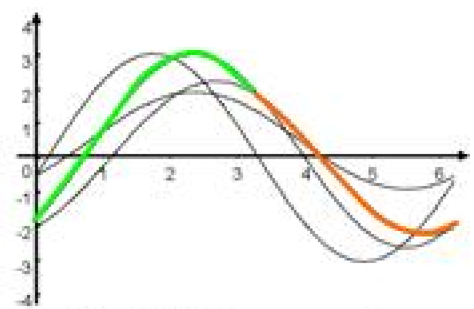
$$d(conv(P),\theta)=\max_{p\in P}\{d(p,\theta)\}$$

Computation of HPR

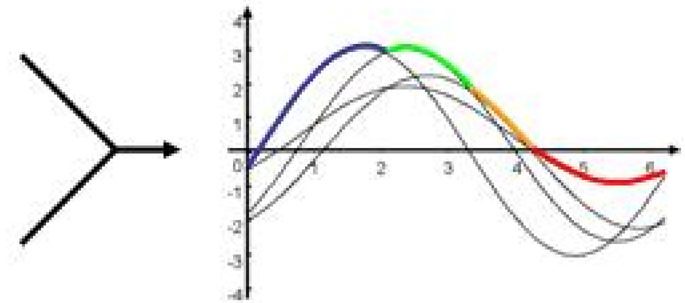




O(n/2) fragments



O(n/2) fragments

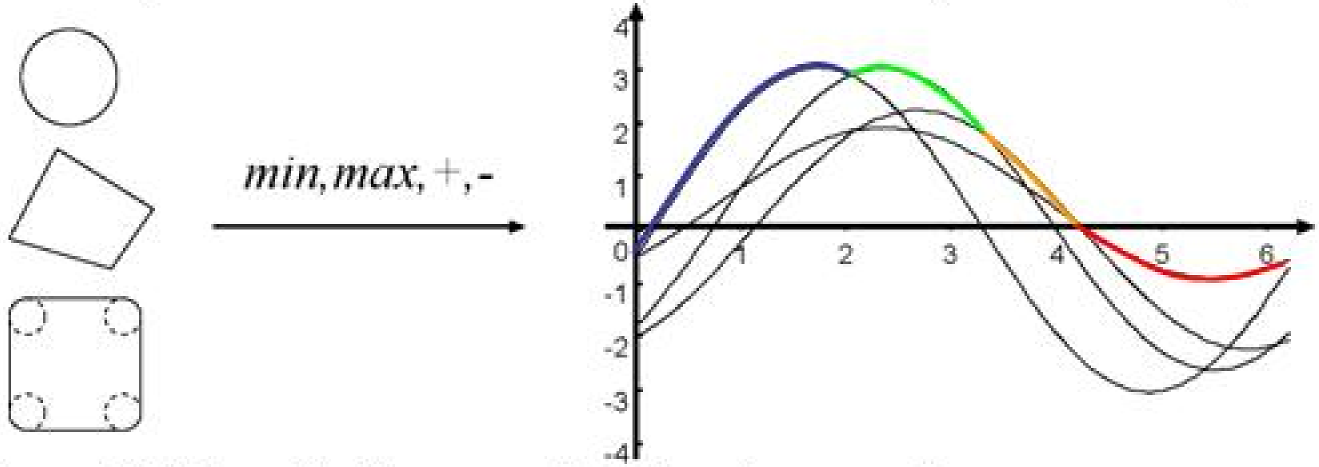


O(n) fragments

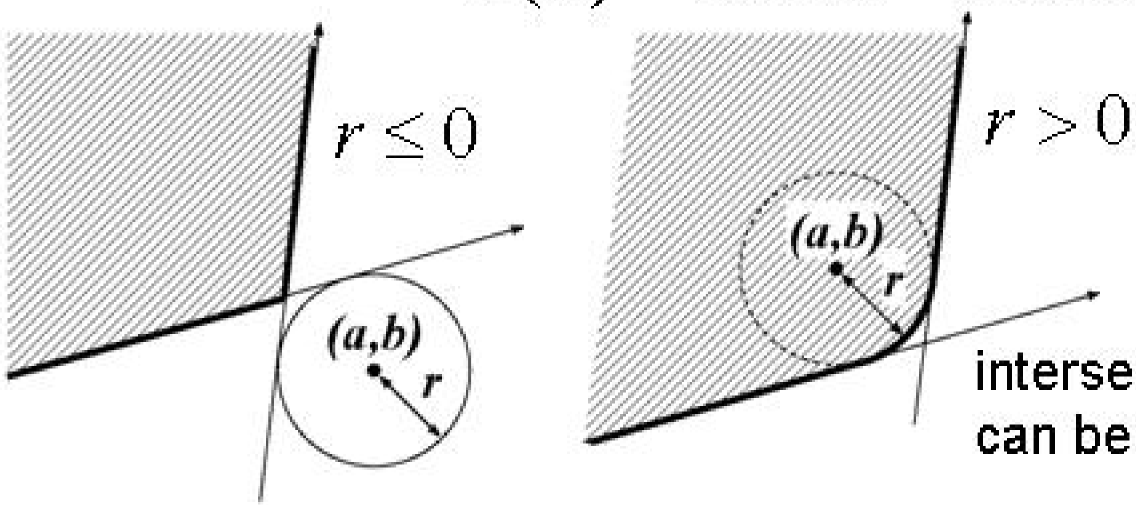
$$\left.\begin{array}{c} O(n) \\ O(n/2)\times 2 \\ O(n/4)\times 4 \\ \vdots \\ O(1)\times n \end{array}\right\} = O(n\log n)$$

Region construction from HPR

- Input objects: circle or (rounded) convex polygon



•Primitive HPR: $d(\theta) = a\sin\theta - b\cos\theta + r$



intersection of *k* primitive regions can be computed in O(k) time.

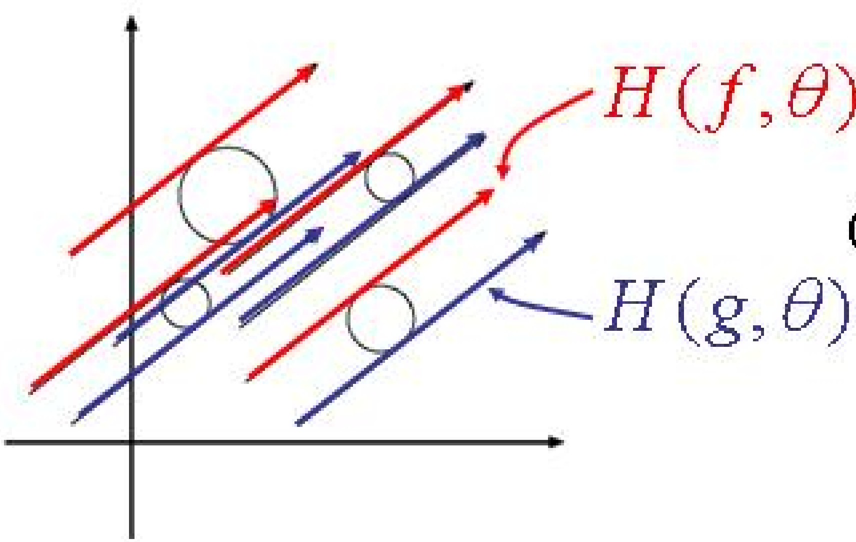
HPR of inner/outer convex hull

- HPR $f(\theta)$ of

$$f(\theta) = \min_{C \in CH(S)} \{d(C,\theta)\} = \max_{\hat{p} \in S} \{-d(\hat{p}, \theta + \pi)\}$$

- HPR $g(\theta)$ of $\overline{conv}(S)$

$$g(\theta) = \max_{C \in CH(S)} \{d(C,\theta)\} = \max_{\hat{p} \in S} \{d(\hat{p}, \theta)\}$$



S: n circles/ convex polygons

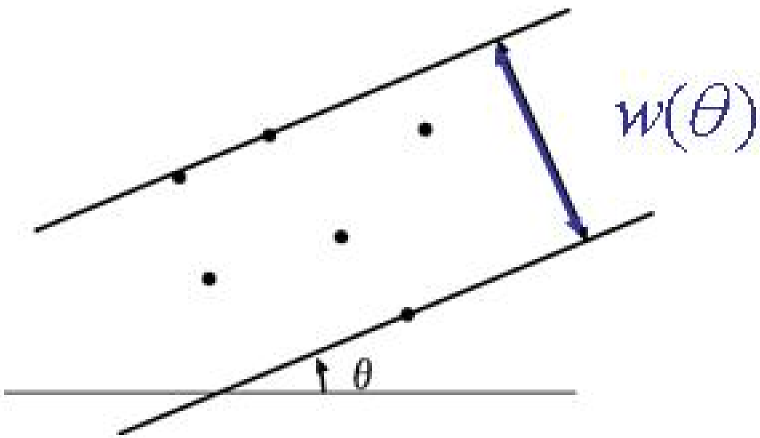
↓

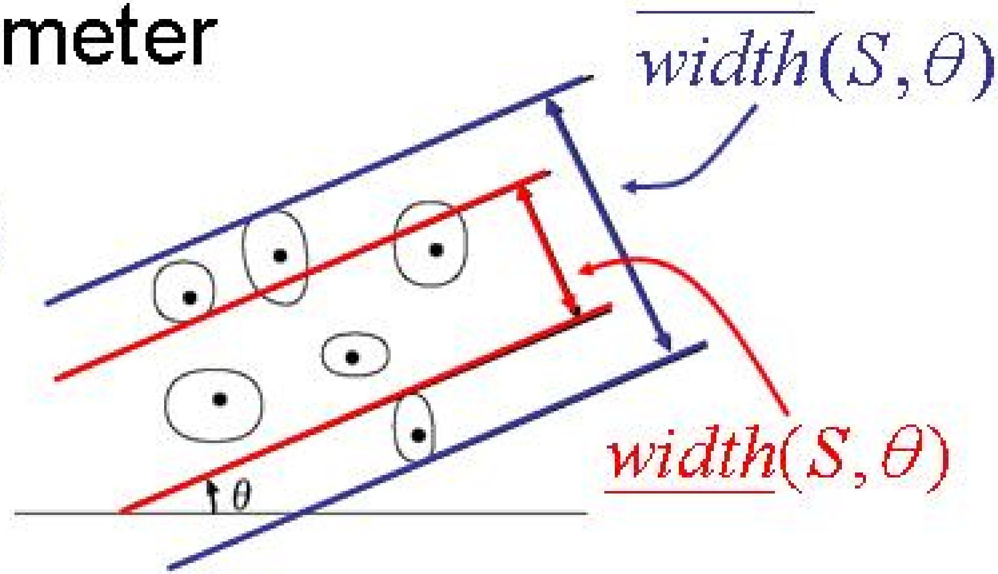
Construction of $\underline{conv}(S)$ $\overline{conv}(S)$

→O(n log n) time.

Diameter of points

- Error bound of diameter



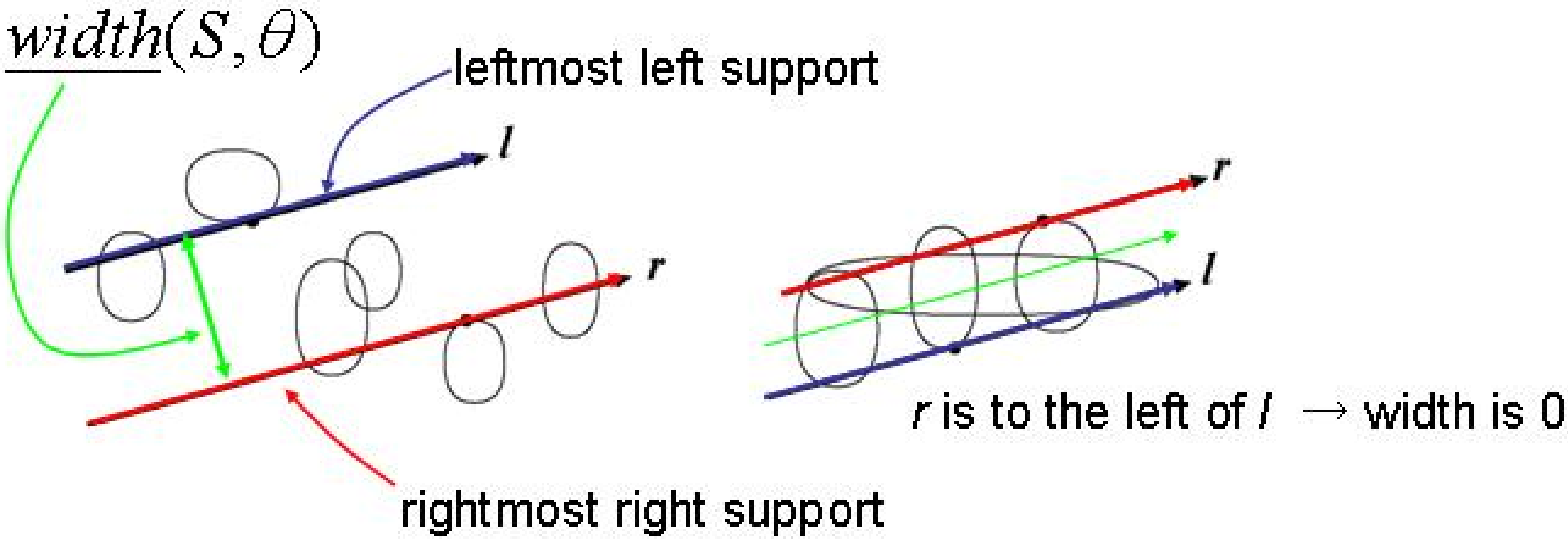


$$diam(P) = \max_{0 \le \theta < 2\pi} \{w(\theta)\}$$

$$\underline{diam}(S) = \max_{0 \le \theta < 2\pi} \{\underline{width}(S,\theta)\}$$

$$\overline{diam}(S) = \max_{0 \le \theta < 2\pi} \{\overline{width}(S,\theta)\}$$

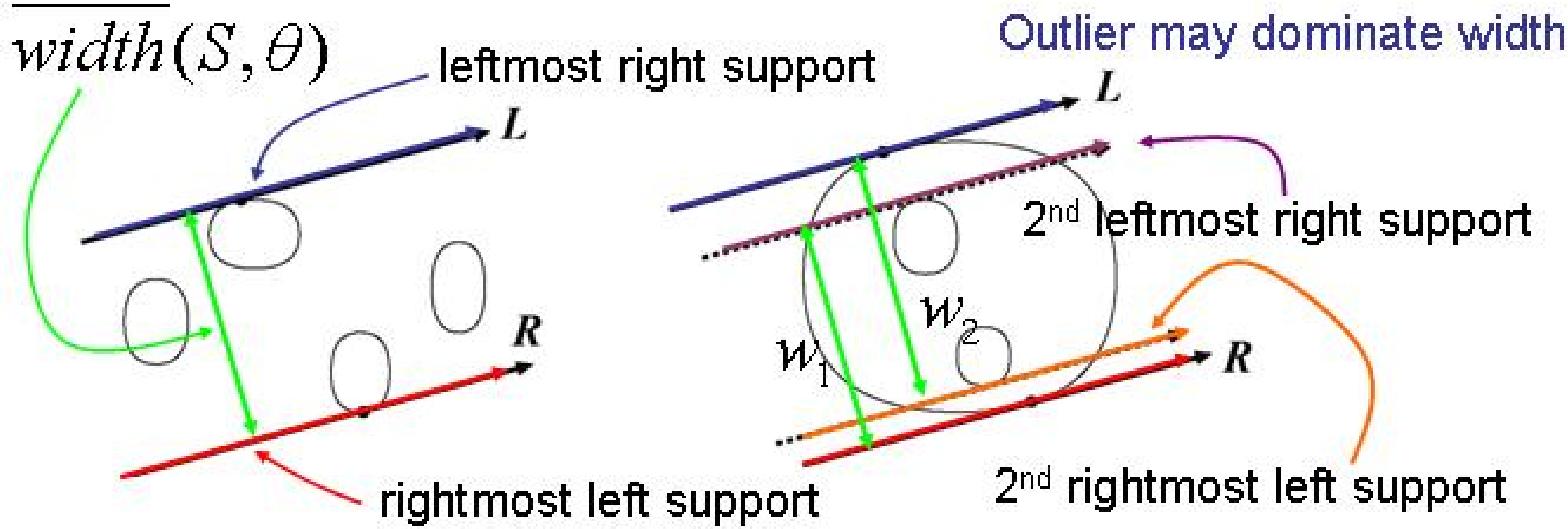
How to compute min. of possible width



$$\underline{width}(S,\theta) = \max\{f(\theta) + f(\theta+\pi), 0\}$$

The min. of possible width can be computed in O(n log n) time

How to compute max. of possible width



$$\overline{width}(S,\theta) = g(\theta) + g(\theta+\pi) \qquad h(\theta) = 2 - \max_{\hat{p}\in S}\{d(\hat{p},\theta)\}$$

or

$$\overline{width}(S,\theta) = \max\left\{\begin{matrix} g(\theta)+h(\theta+\pi), \\ h(\theta)+g(\theta+\pi) \end{matrix}\right\}$$

The max. of possible width can be computed in $O(n\cdot\alpha(n)\cdot\log n)$ time

Conclusion

- Geometric algorithms with imprecise input
  - Points are known only up to some accuracy.
- Half-plane representation of a convex region
- **Accuracy guaranteed solutions from HPR**