Voronoi Diagrams

Computational Geometry [csci 3250]
Laura Toma
Bowdoin College

Outline

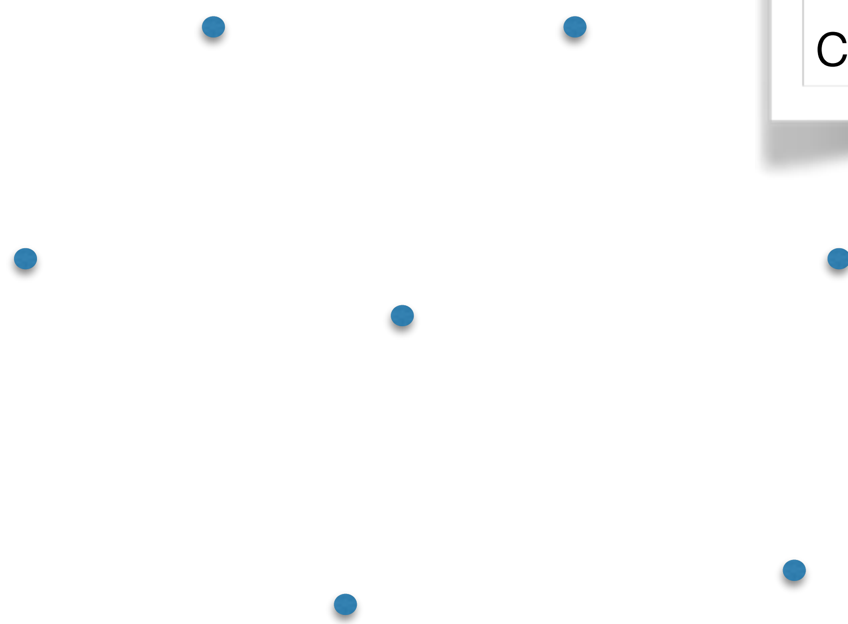
- Voronoi diagrams in 2D
  - Definition
  - Properties
  - Algorithms
  - Applications
  - Extensions
- Delaunay triangulations (next time)

- Reading: O'Rourke chapter 5

Voronoi Diagram Vor(P)

Let P = {p_1, p_2,…,p_n} a set of n points in the plane (called **sites**)

We want to subdivide space according to which site is closest.

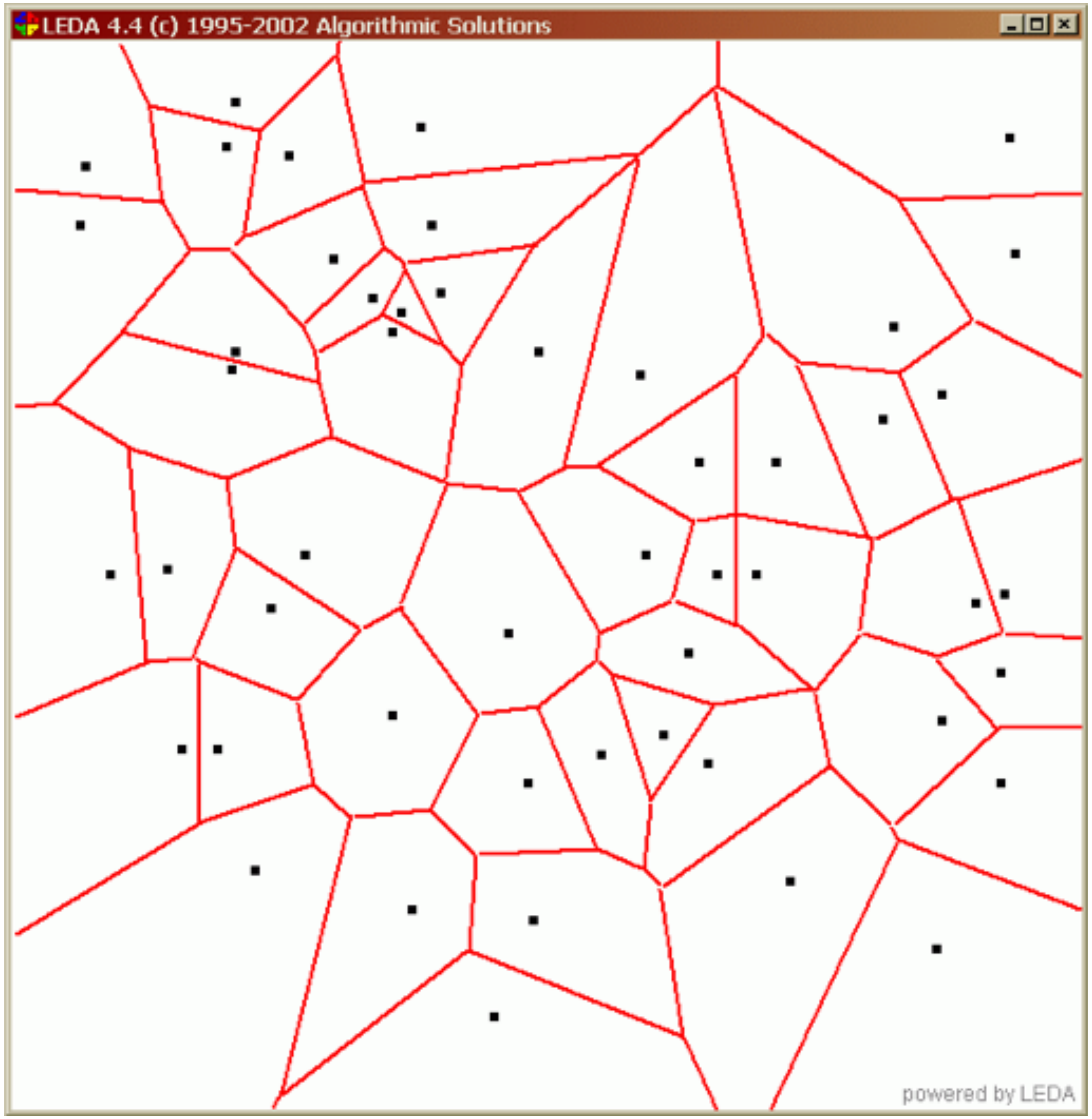


Old! Concept discussed in 1850 by Dirichelet, paper in 1908 by Voronoi

Voronoi Diagram

- n=2

In general



Voronoi Diagram Vor(P)

Let $P = \{p_1, p_2, \ldots, p_n\}$ a set of n points in the plane (called **sites**)

- The Voronoi cell of p_i is a region in the plane defined as

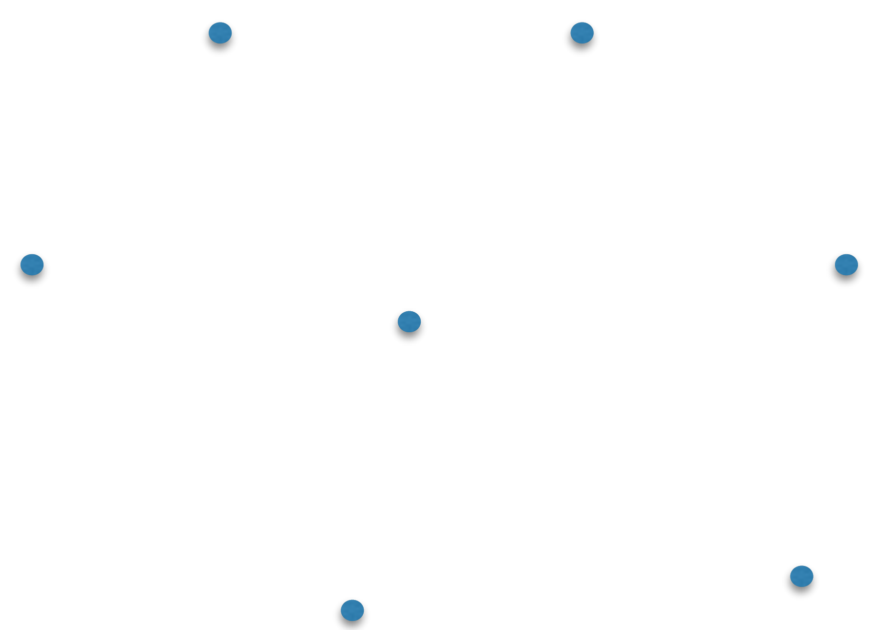
  Vor(p_i): all points in the plane that are closer to p_i than to any other site

  $Vor(p_i) = \{ q \mid ||p_i q|| <= ||p_j q||, \text{for any } j \ != i\}$

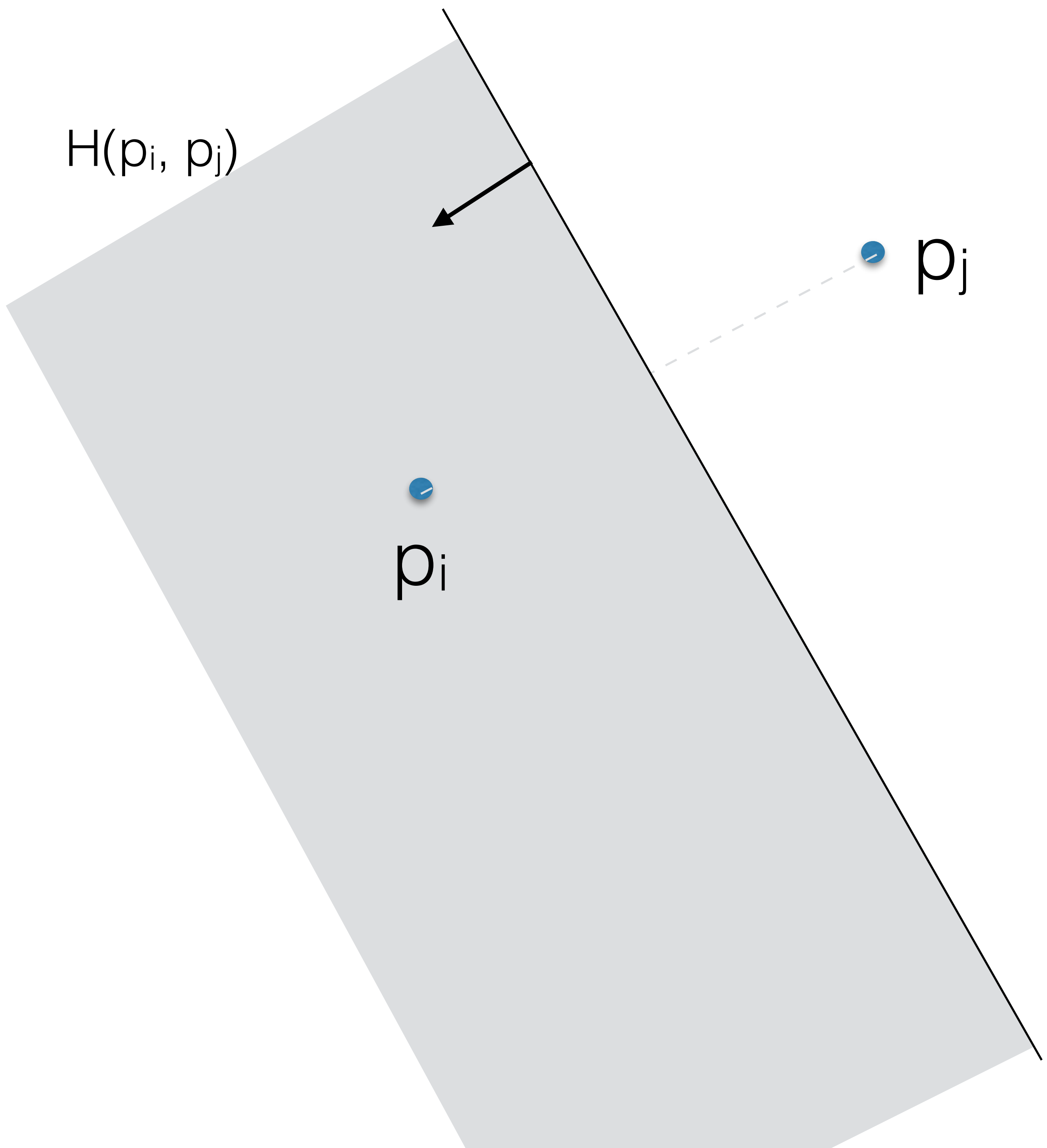
- The Voronoi diagram of P : $Vor(P) = U\ Vor(p_i)$

- Vor(P) defines a partition of the plane
  - for any point q in the plane, let p be its nearest site. Then q belongs to the Voronoi cell of p

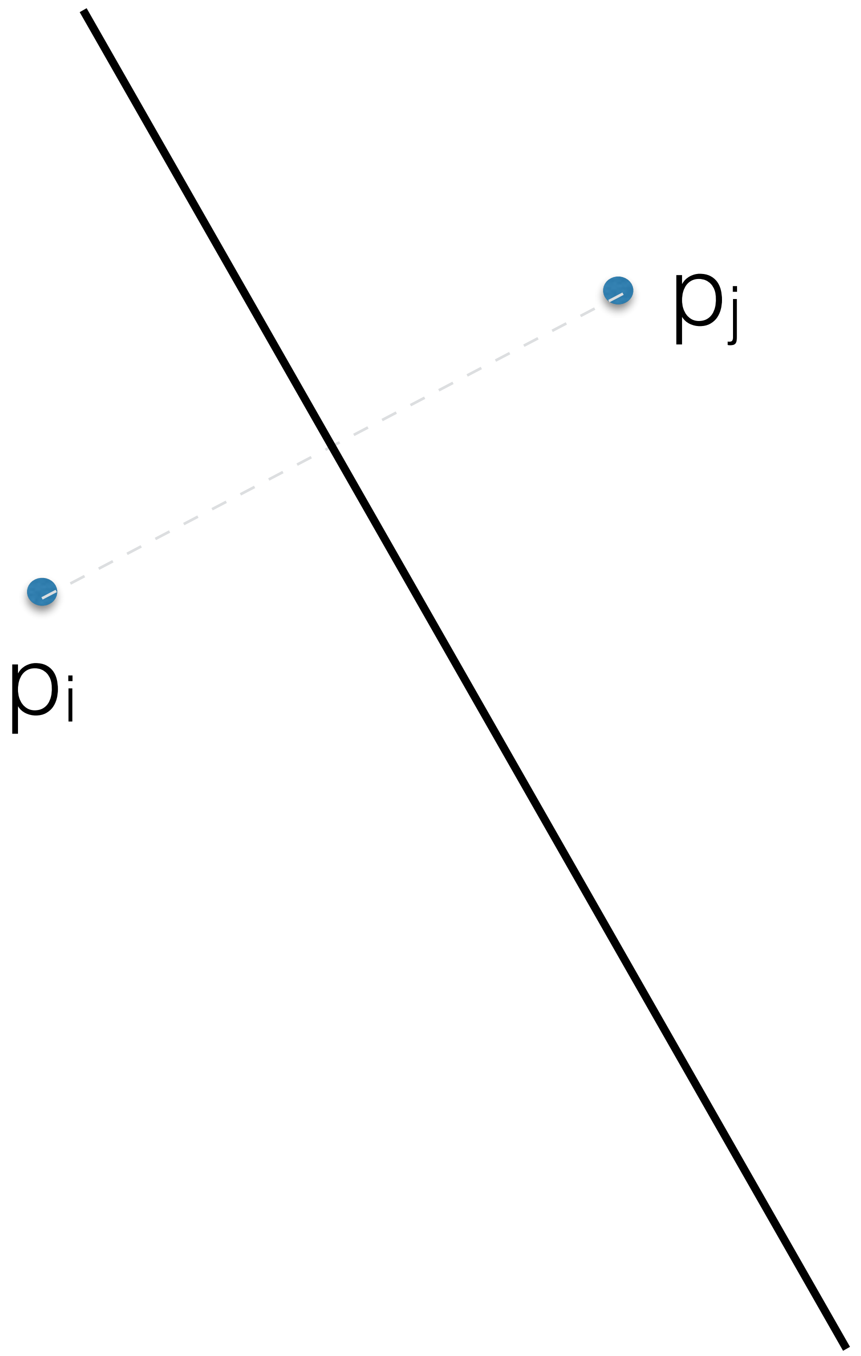
The problem: Given $P = \{p_1, p_2, \ldots, p_n\}$, compute Vor(P)



Given two points p_i and p_j, the set of points that are strictly closer to p_i than to p_j is the open **halfplane** bounded by the perpendicular bisector. Denote it $H(p_i, p_j)$

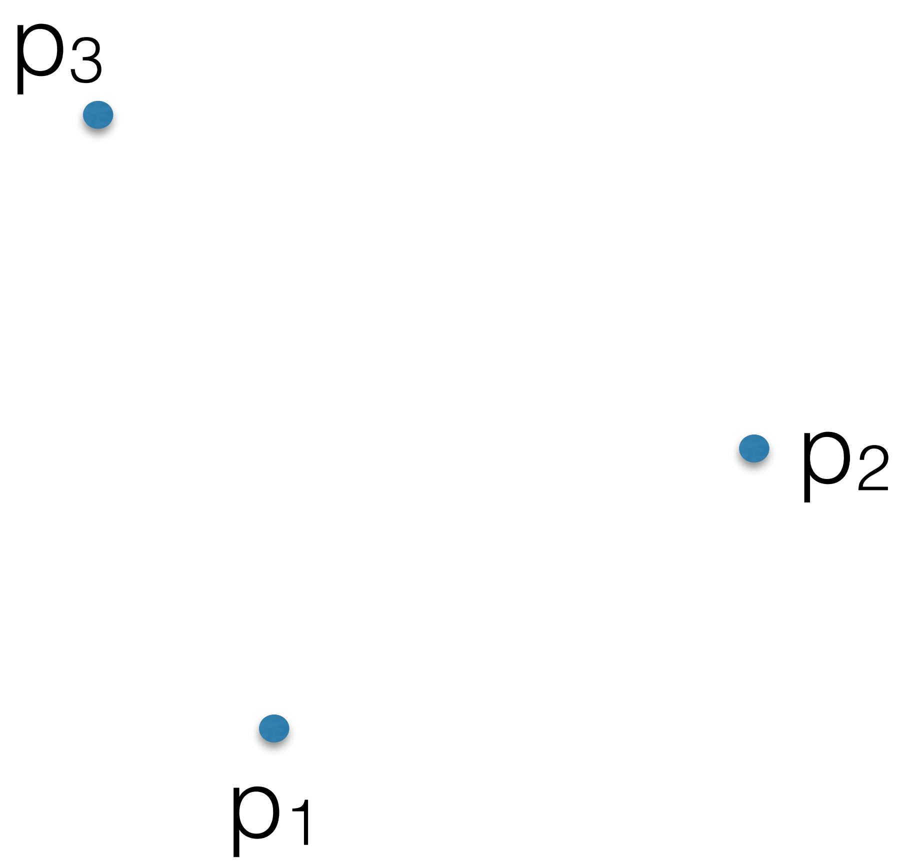


Voronoi Diagram



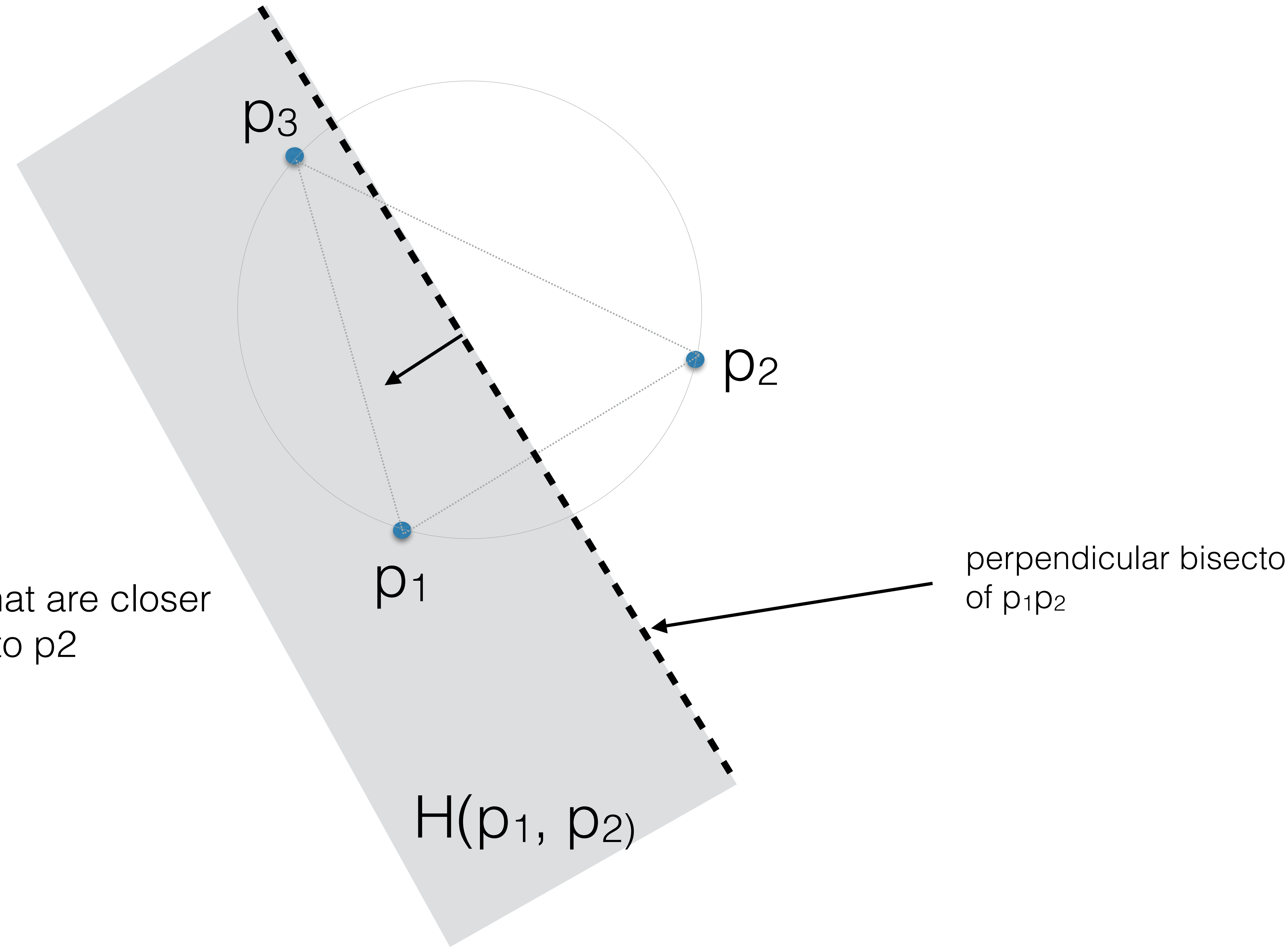
Voronoi Diagram

- n=3



Voronoi Diagram

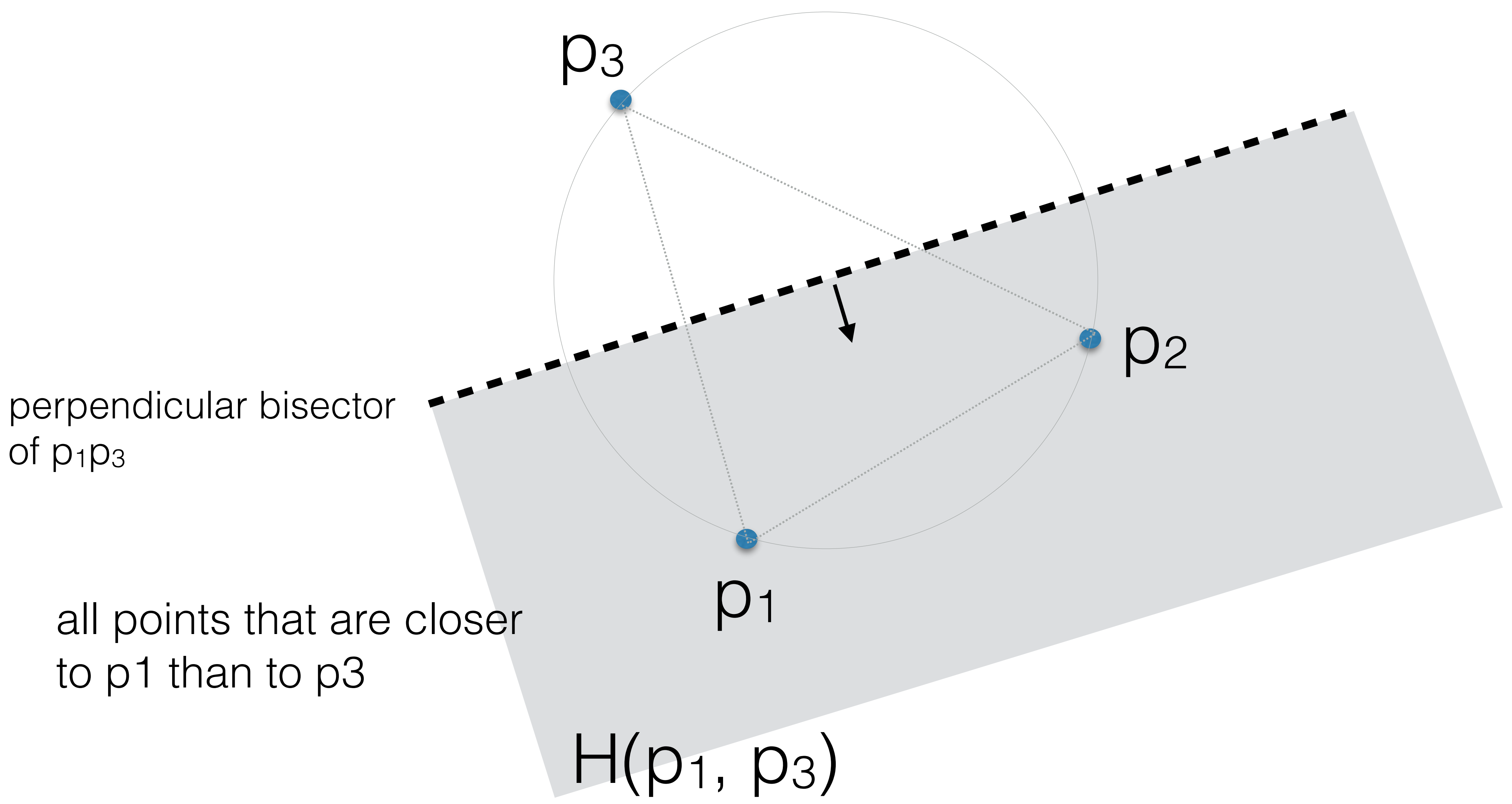
- n=3



all points that are closer
to p1 than to p2

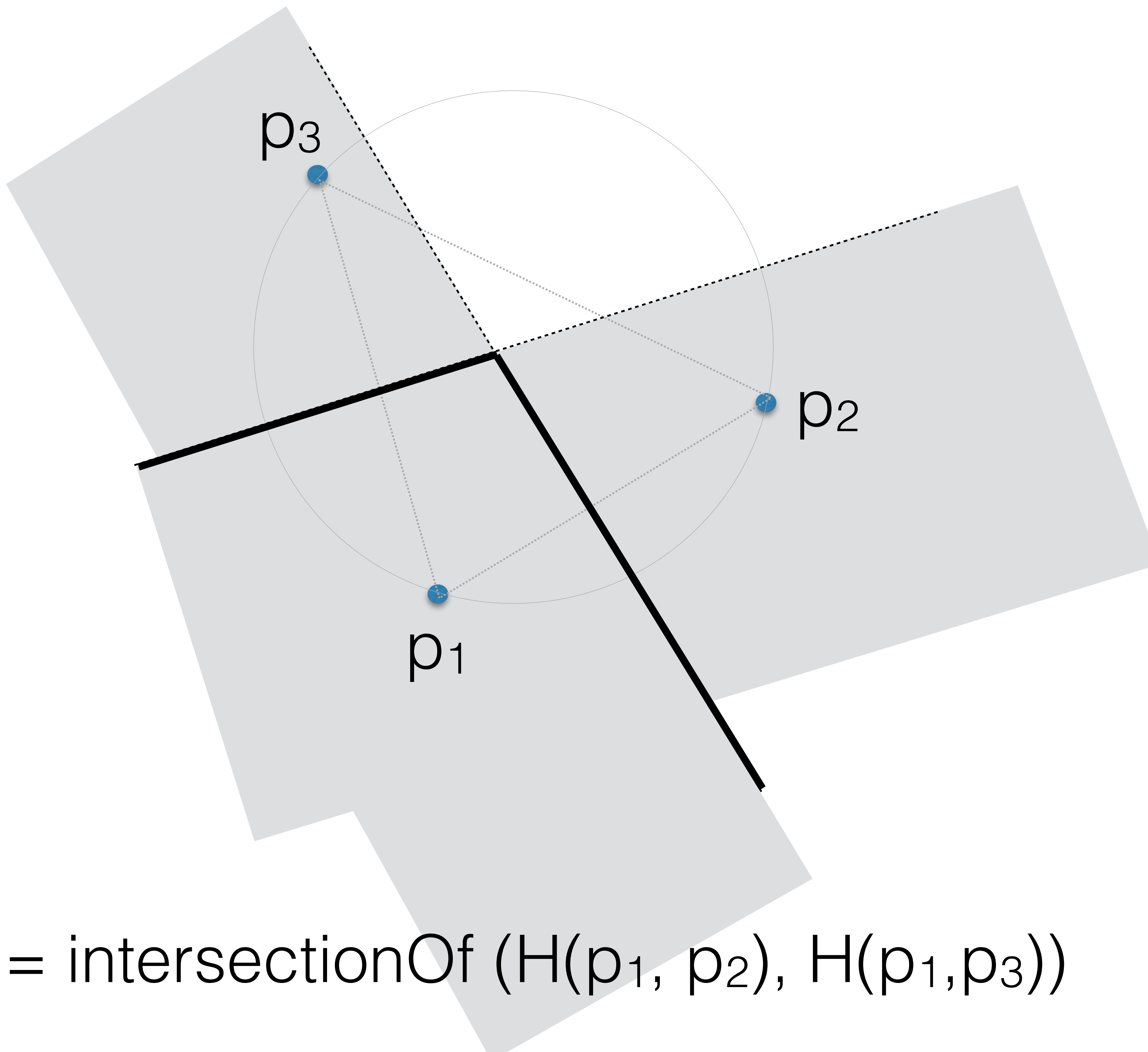
Voronoi Diagram

- n=3



Voronoi Diagram

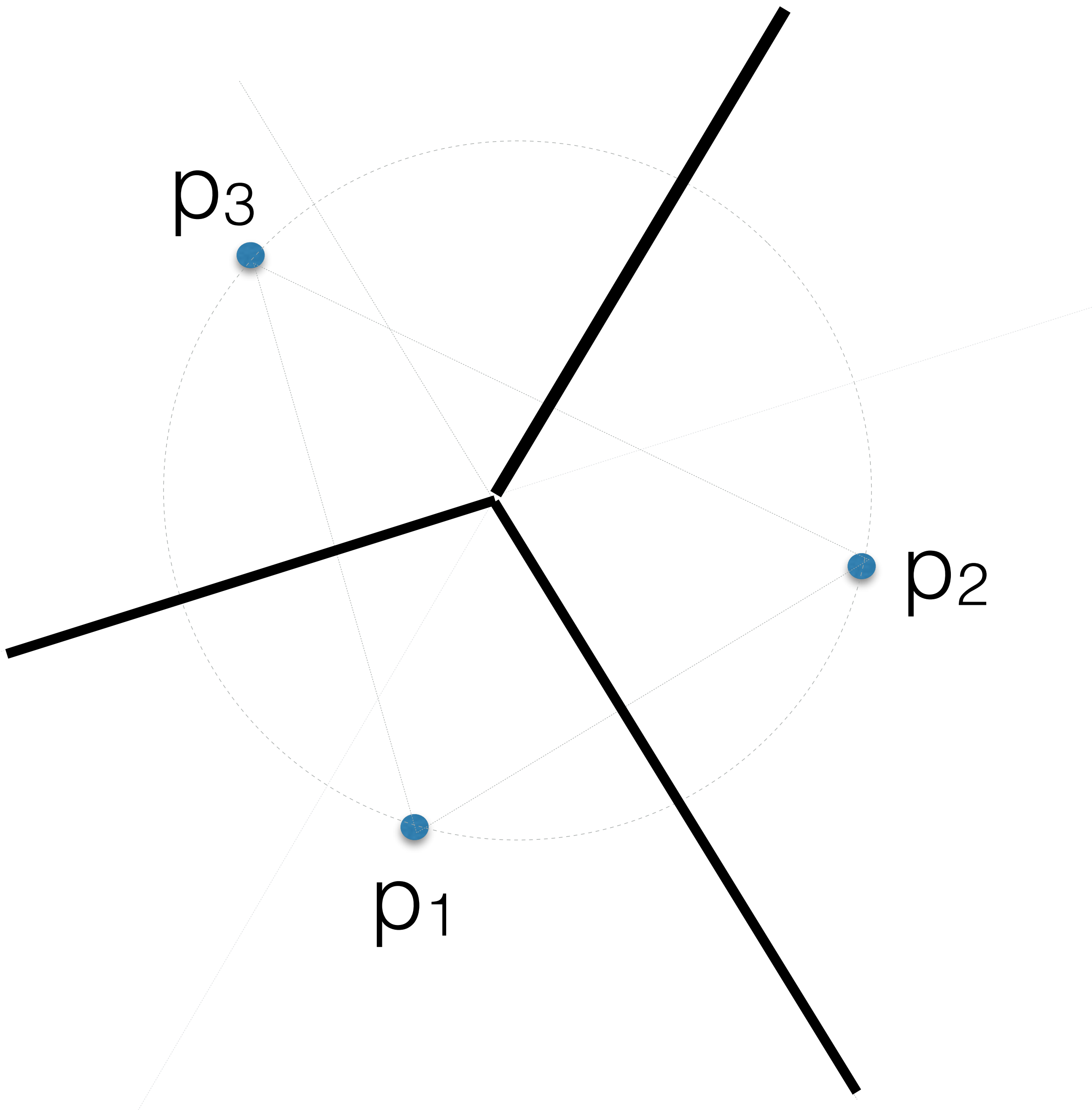
- n=3



Vor(p_1) = intersectionOf (H(p_1, p_2), H(p_1,p_3))

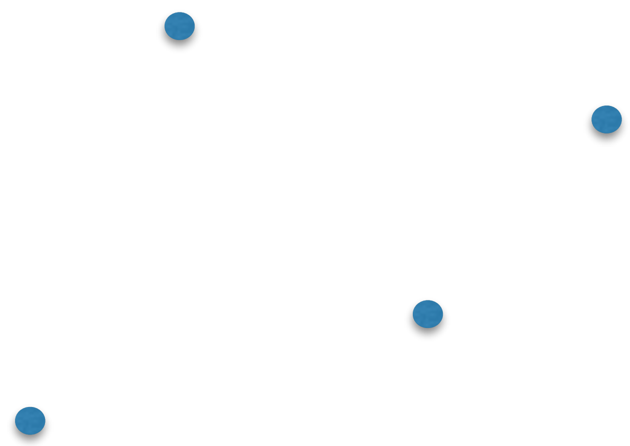
Voronoi Diagram

- n=3



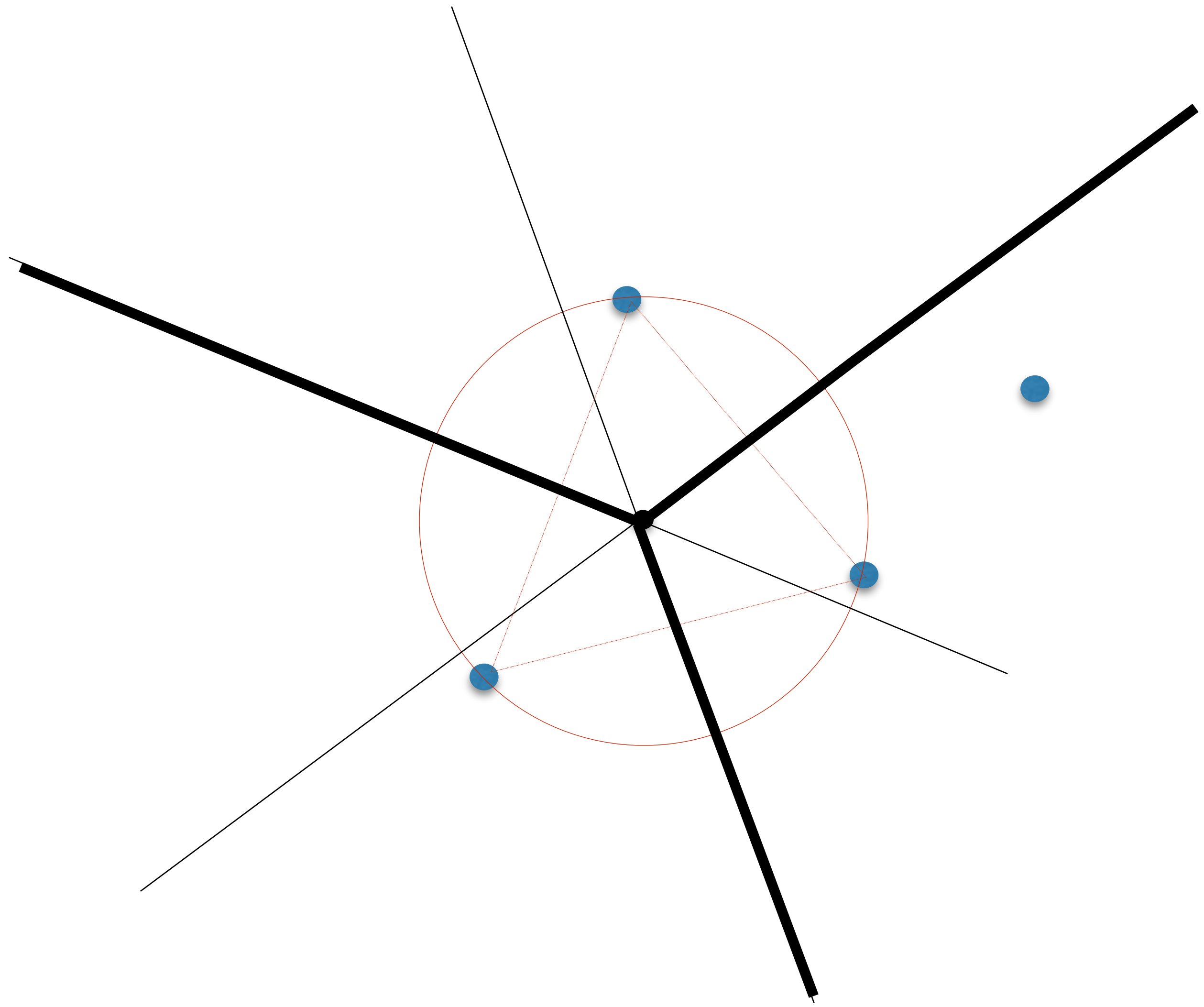
Voronoi Diagram

- n=4



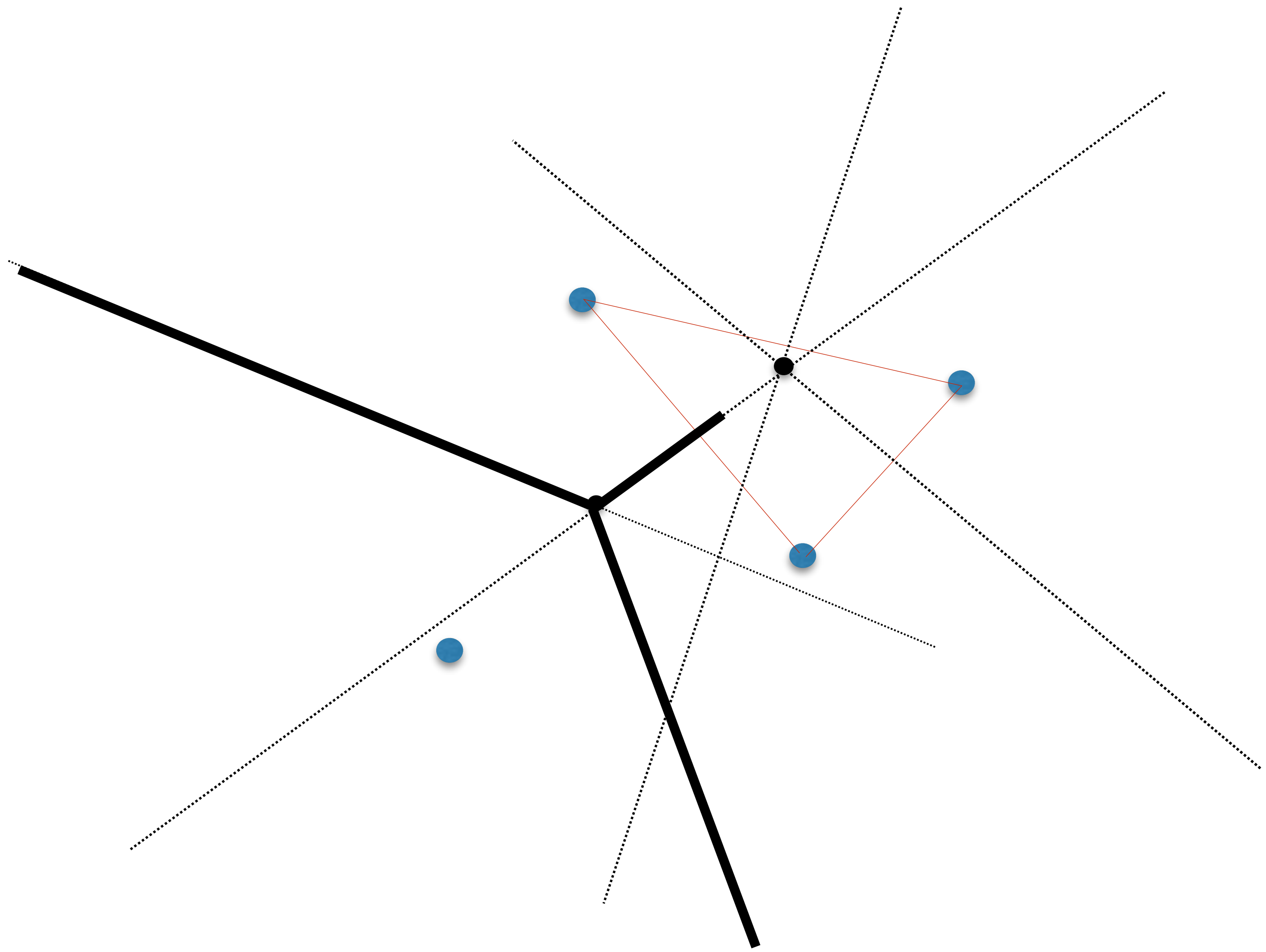
Voronoi Diagram

- n=4



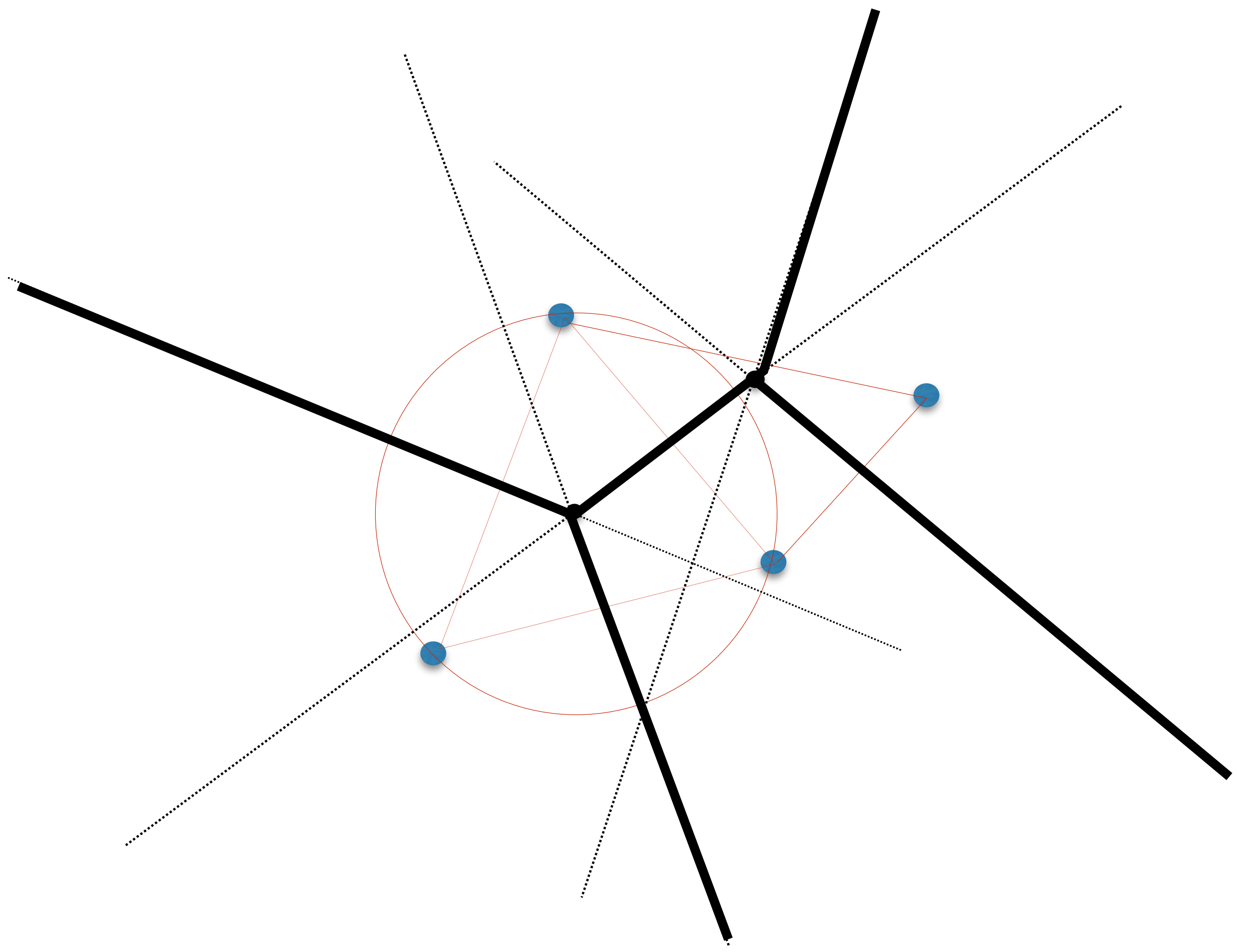
Voronoi Diagram

- n=4



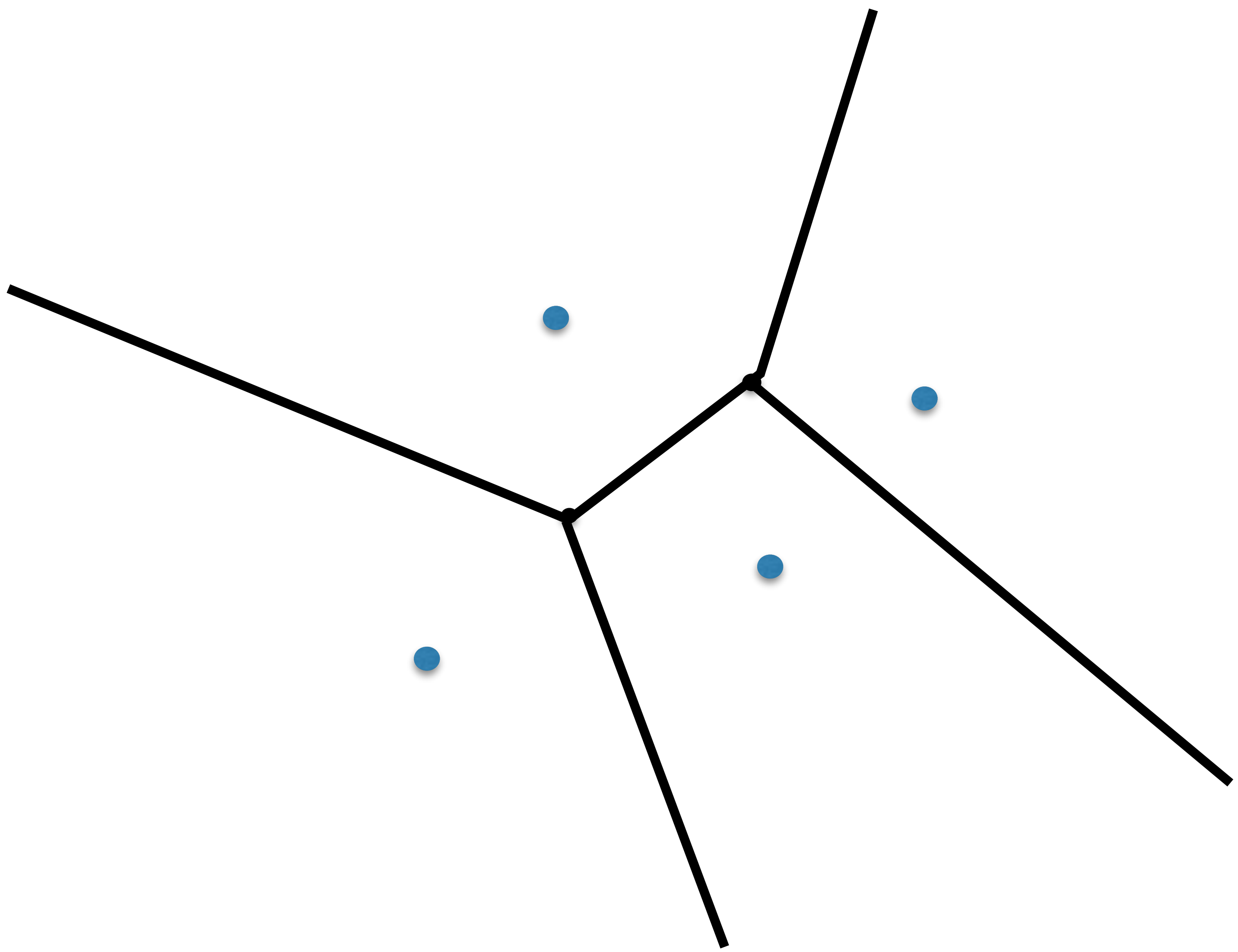
Voronoi Diagram

- n=4



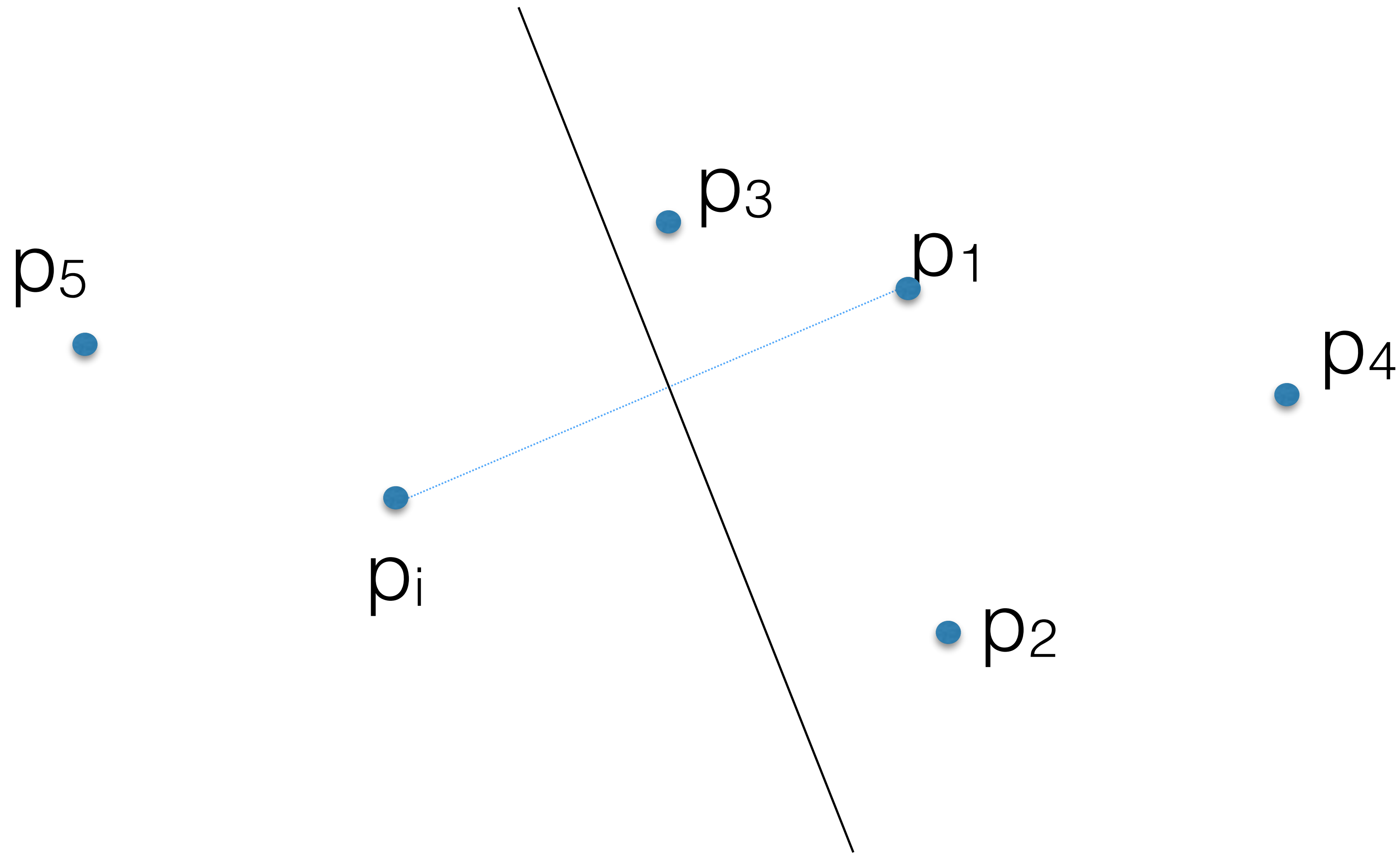
Voronoi Diagram

- n=4



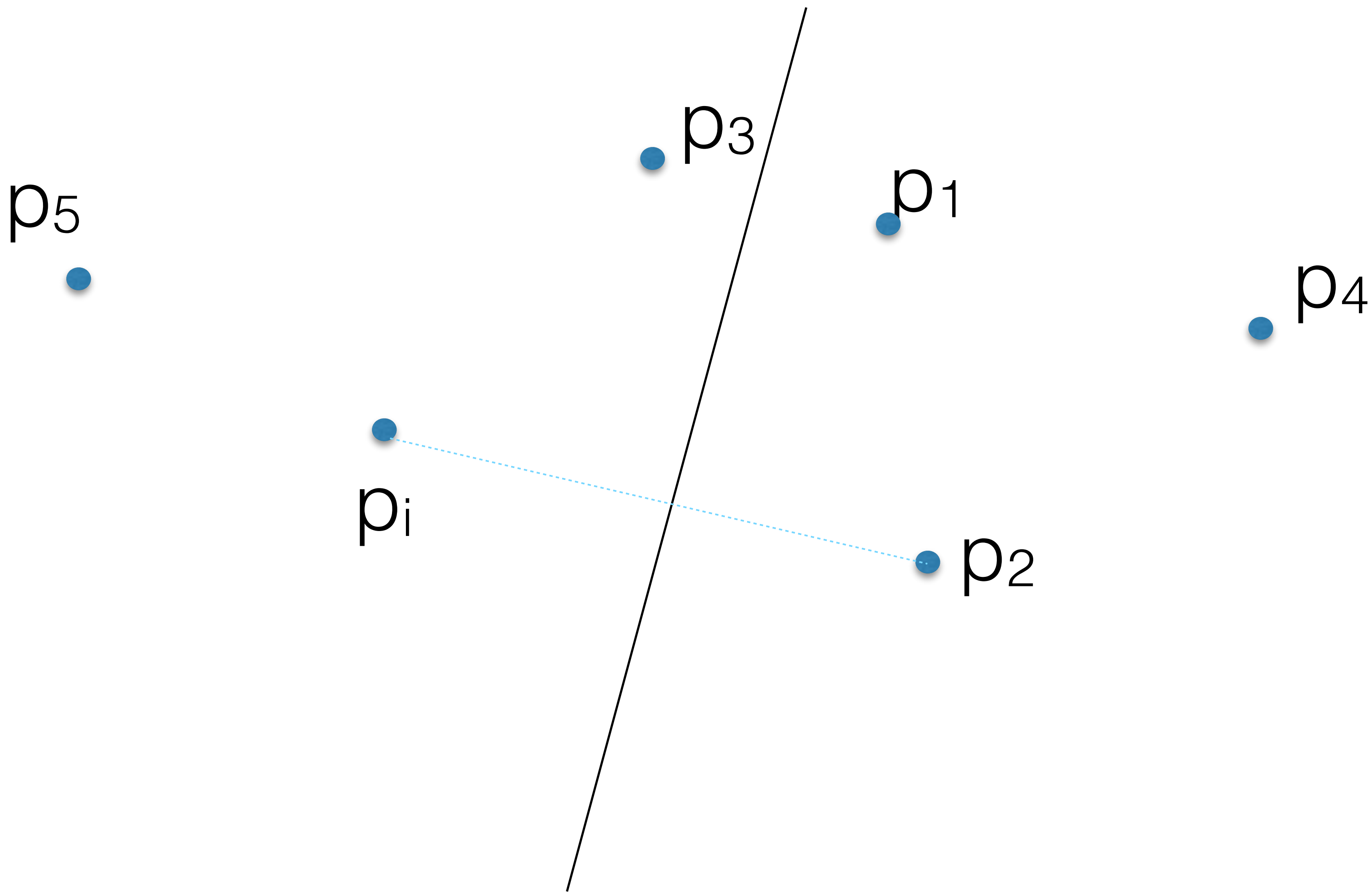
Vor(P) as Intersection of Halfplanes

- A point lies in Vor(pi) if and only if it lies in the intersection of $H(p_i, p_j)$ for all j (j != i)
- Vor (p_i) = IntersectionOf { $H(p_i, p_j)$, all j != i}



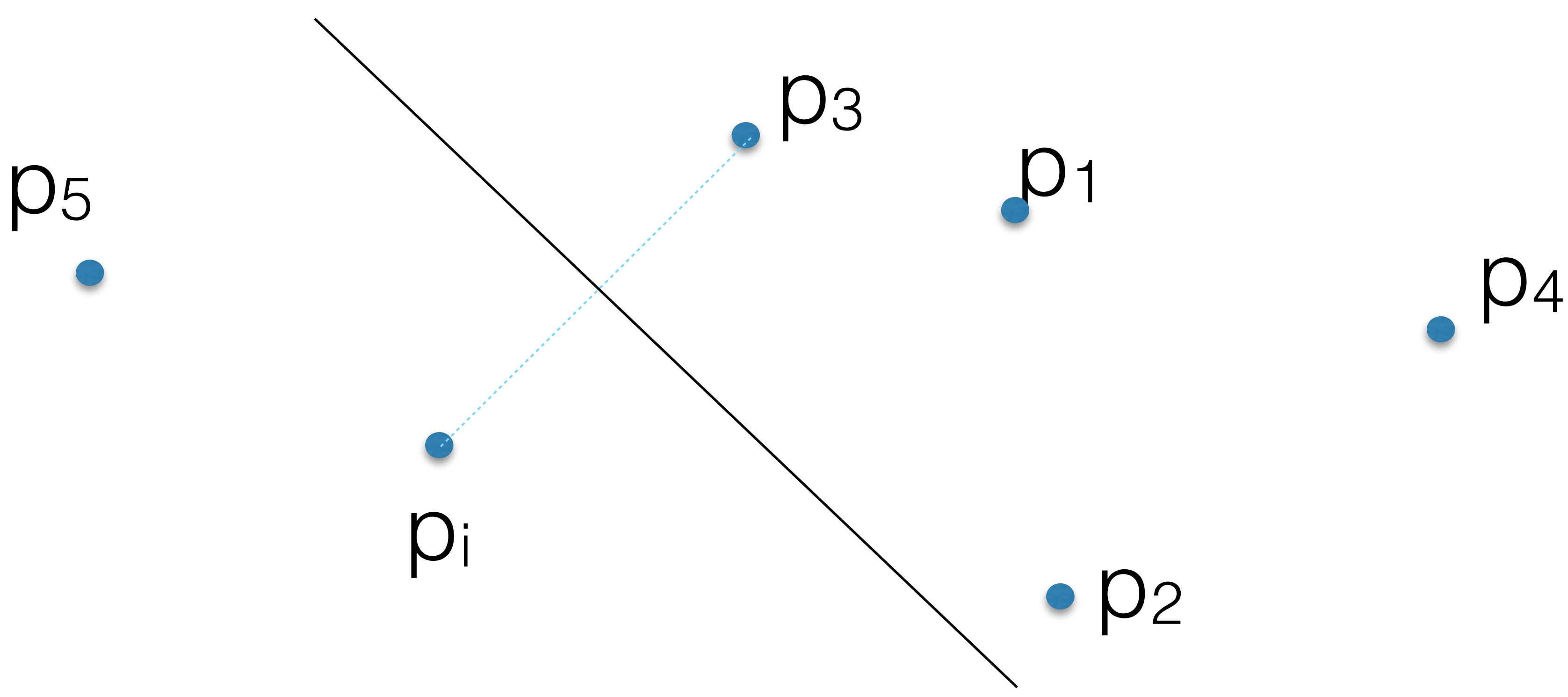
Vor(P) as Intersection of Halfplanes

- A point lies in Vor(pi) if and only if it lies in the intersection of $H(p_i, p_j)$ for all j
- Vor (p_i) = IntersectionOf { $H(p_i, p_j)$, all j != i}



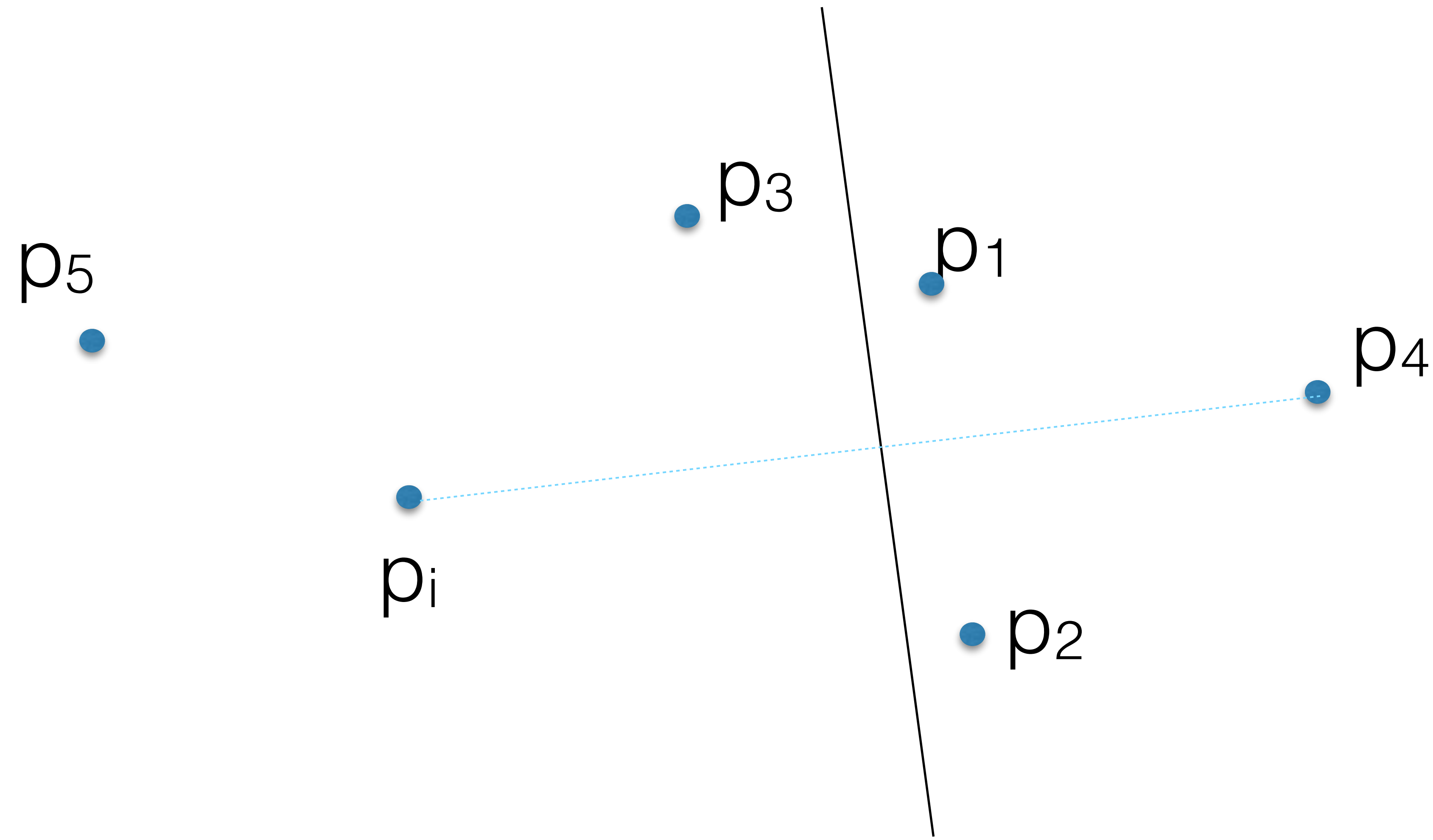
Vor(P) as Intersection of Halfplanes

- A point lies in Vor(pi) if and only if it lies in the intersection of $H(p_i, p_j)$ for all j
- Vor (p_i) = IntersectionOf { $H(p_i, p_j)$, all j != i}



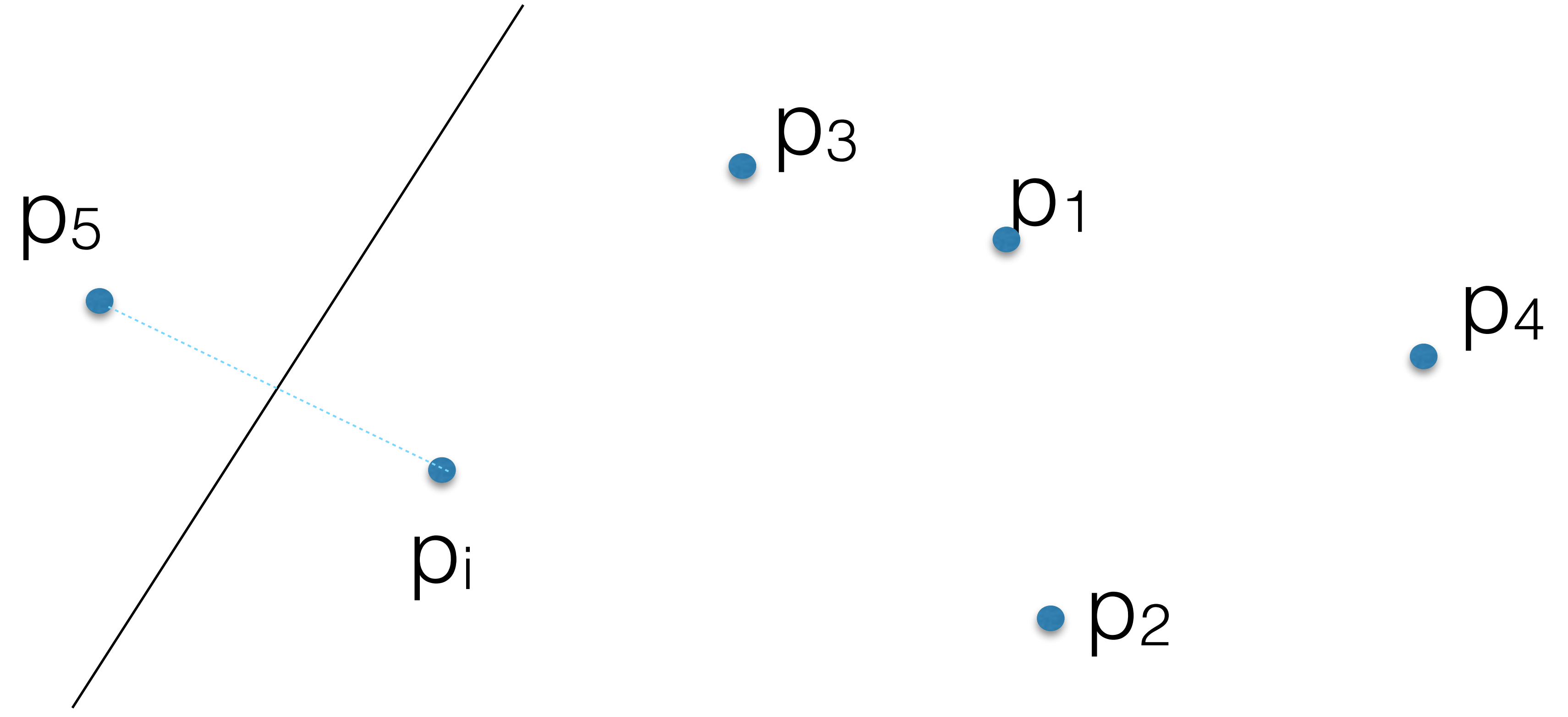
Vor(P) as Intersection of Halfplanes

- A point lies in Vor(pi) if and only if it lies in the intersection of $H(p_i, p_j)$ for all j
- Vor (p_i) = IntersectionOf { $H(p_i, p_j)$, all j != i}



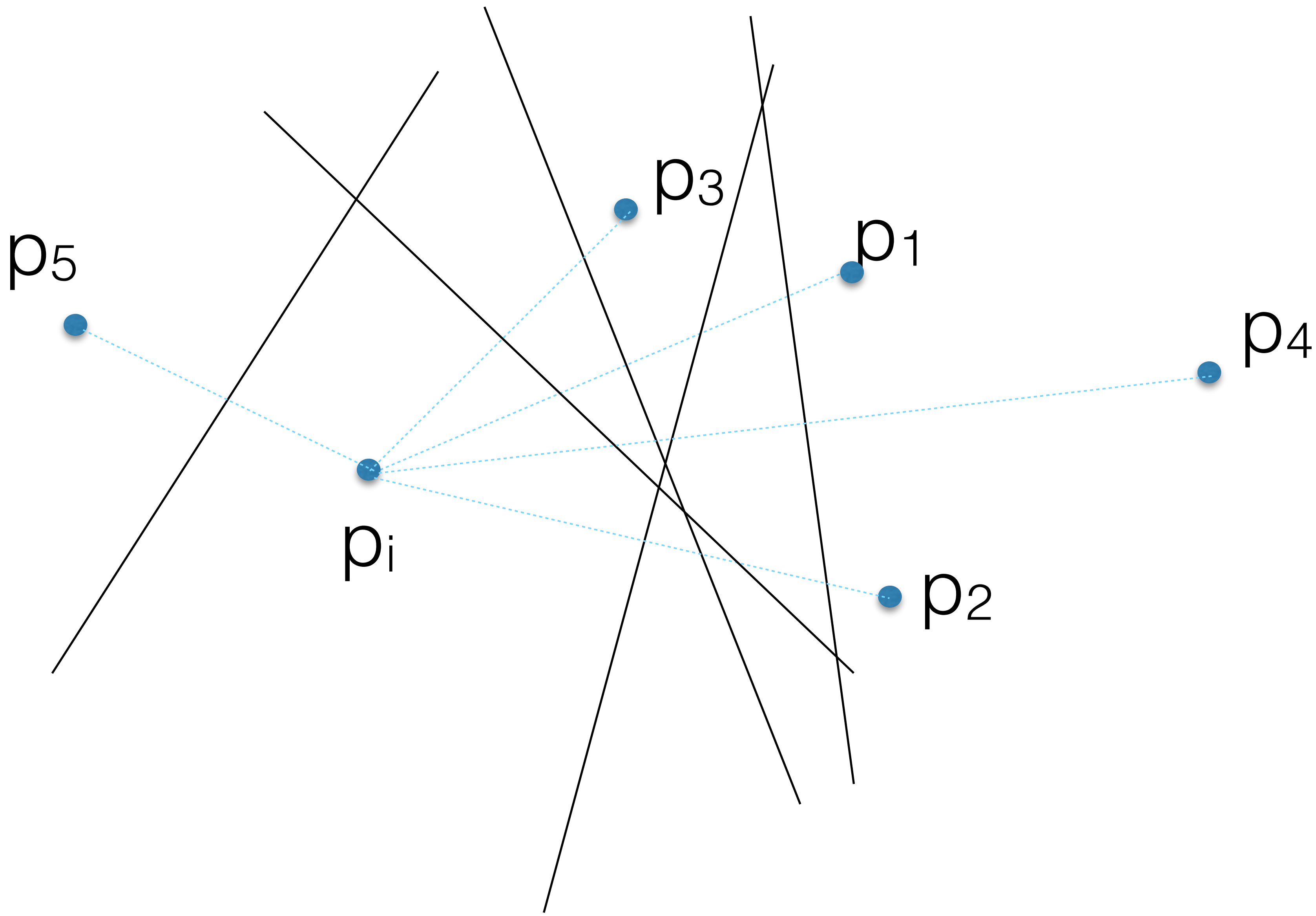
Vor(P) as Intersection of Halfplanes

- A point lies in Vor(pi) if and only if it lies in the intersection of $H(p_i, p_j)$ for all j
- Vor (p_i) = IntersectionOf { $H(p_i, p_j)$, all j != i}



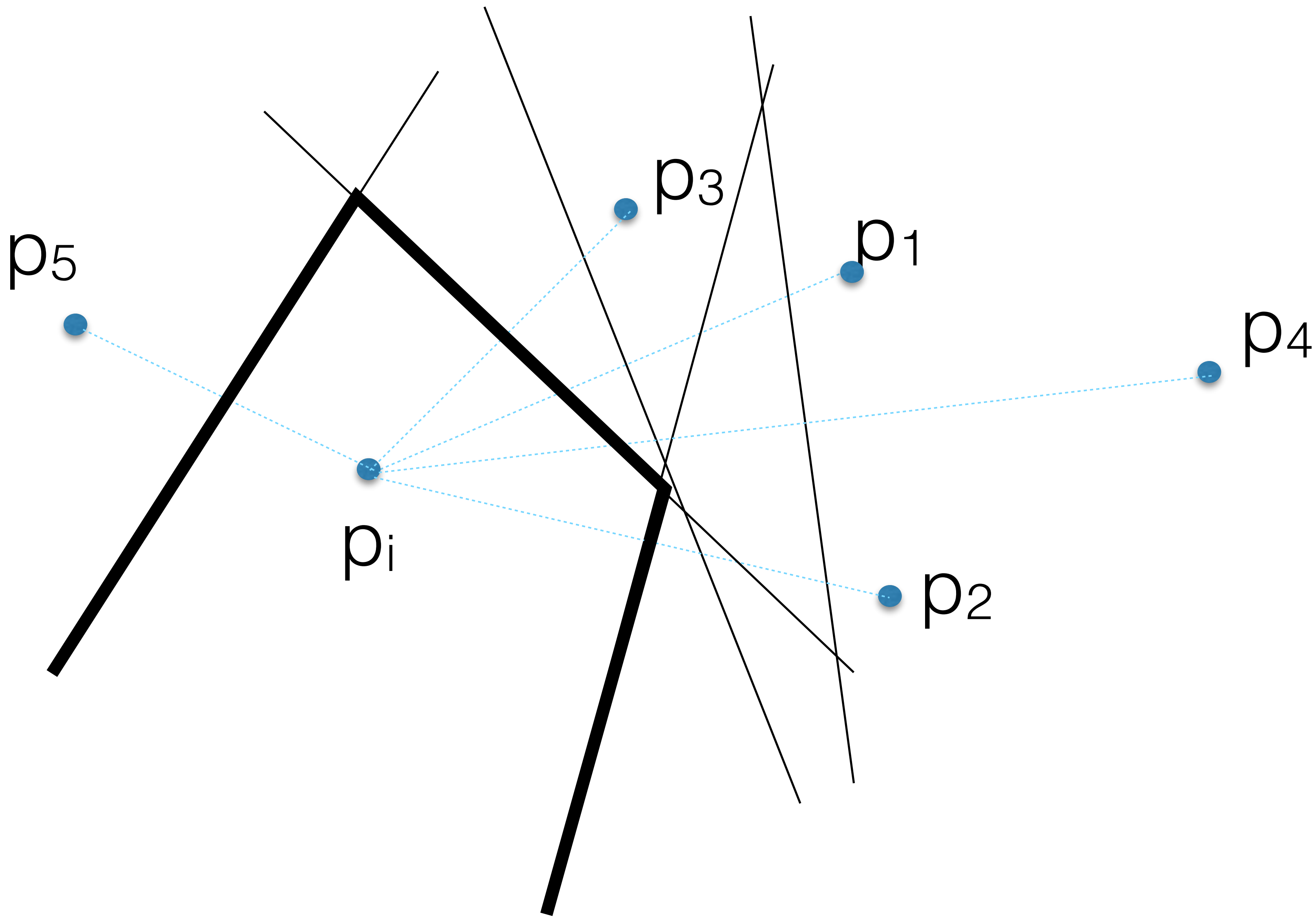
Vor(P) as Intersection of Halfplanes

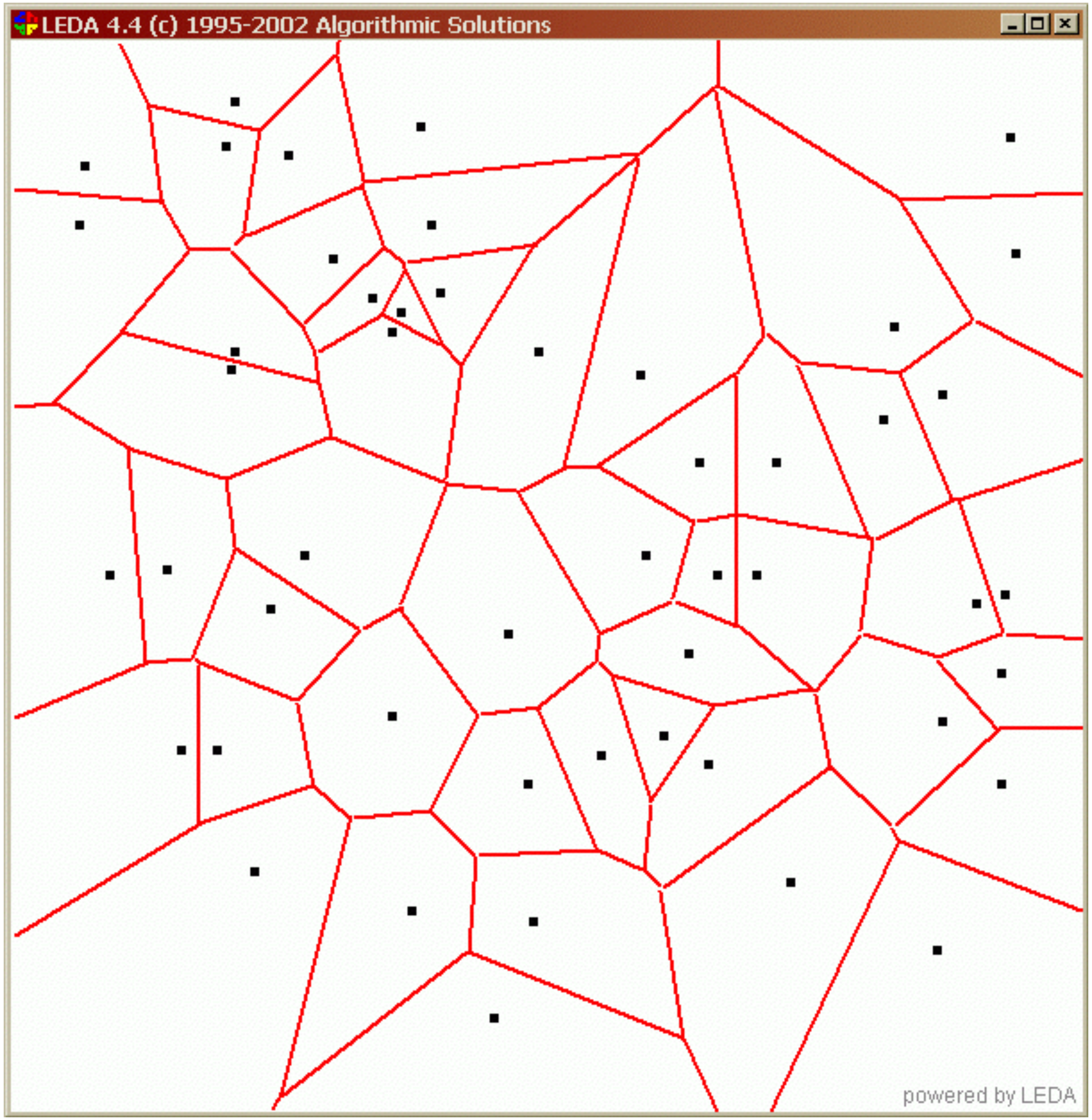
- A point lies in Vor(pi) if and only if it lies in the intersection of $H(p_i, p_j)$ for all j
- Vor (p_i) = IntersectionOf { $H(p_i, p_j)$, all j != i}



Vor(P) as Intersection of Halfplanes

- A point lies in Vor(pi) if and only if it lies in the intersection of $H(p_i, p_j)$ for all j
- Vor (p_i) = IntersectionOf { $H(p_i, p_j)$, all j != i}



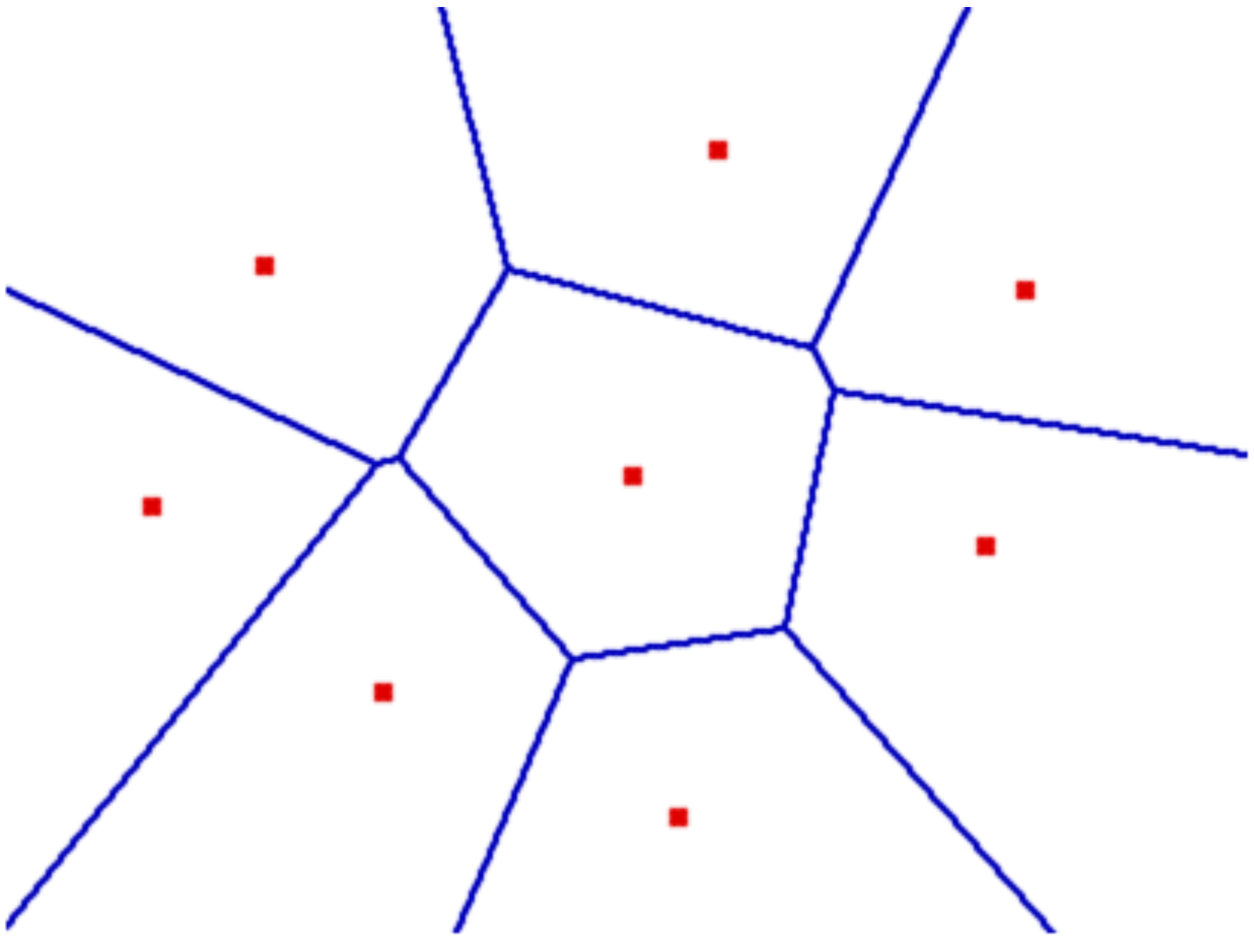

LEDA 4.4 (c) 1995-2002 Algorithmic Solutions
powered by LEDA

Properties of Voronoi Diagram

Properties of Voronoi Diagram

Let $P = \{p_1, p_2, \ldots, p_n\}$ set of points in the plane.

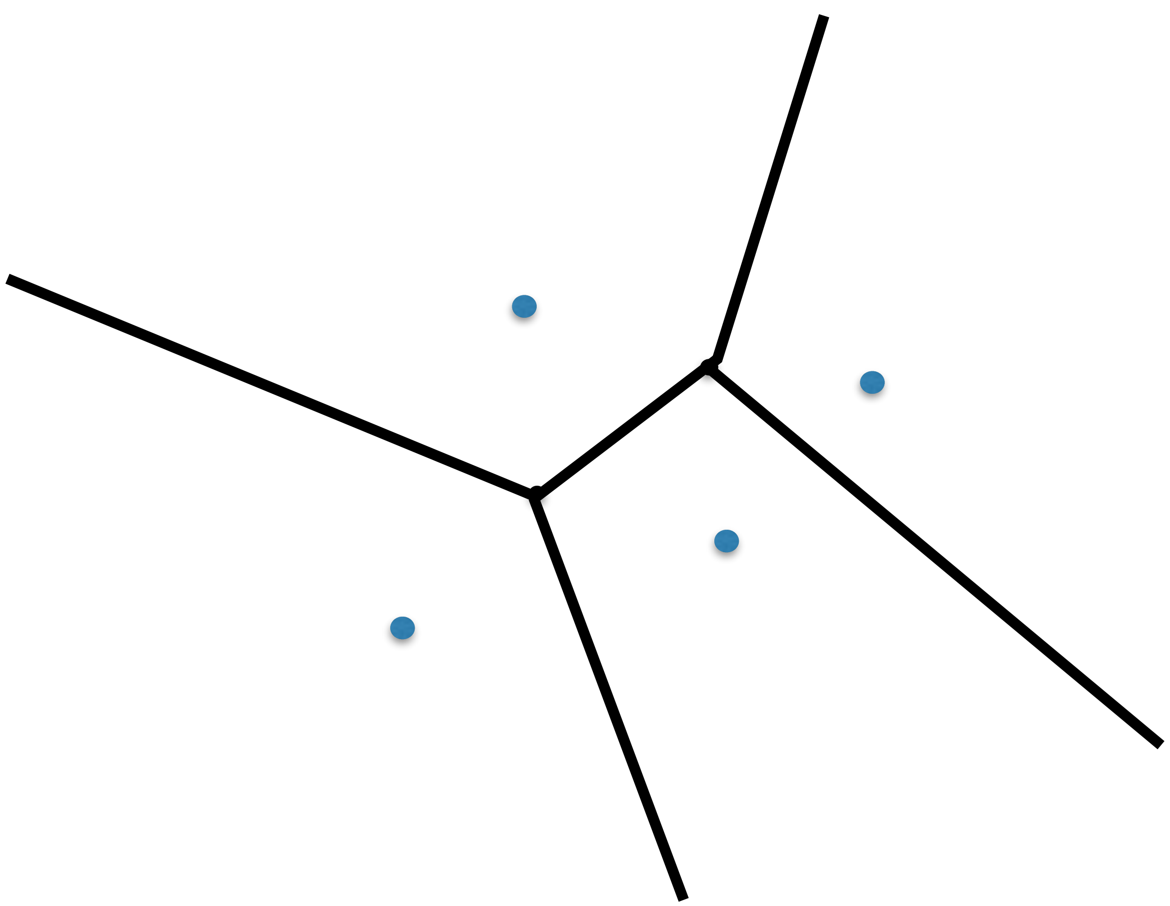
- Vor(P) consists of convex polygons
  - Each cell is intersection of halfplanes, which are convex. Intersection of convex regions is convex.



Properties of Voronoi Diagram

Let $P = \{p_1, p_2, \ldots, p_n\}$ set of points in the plane.

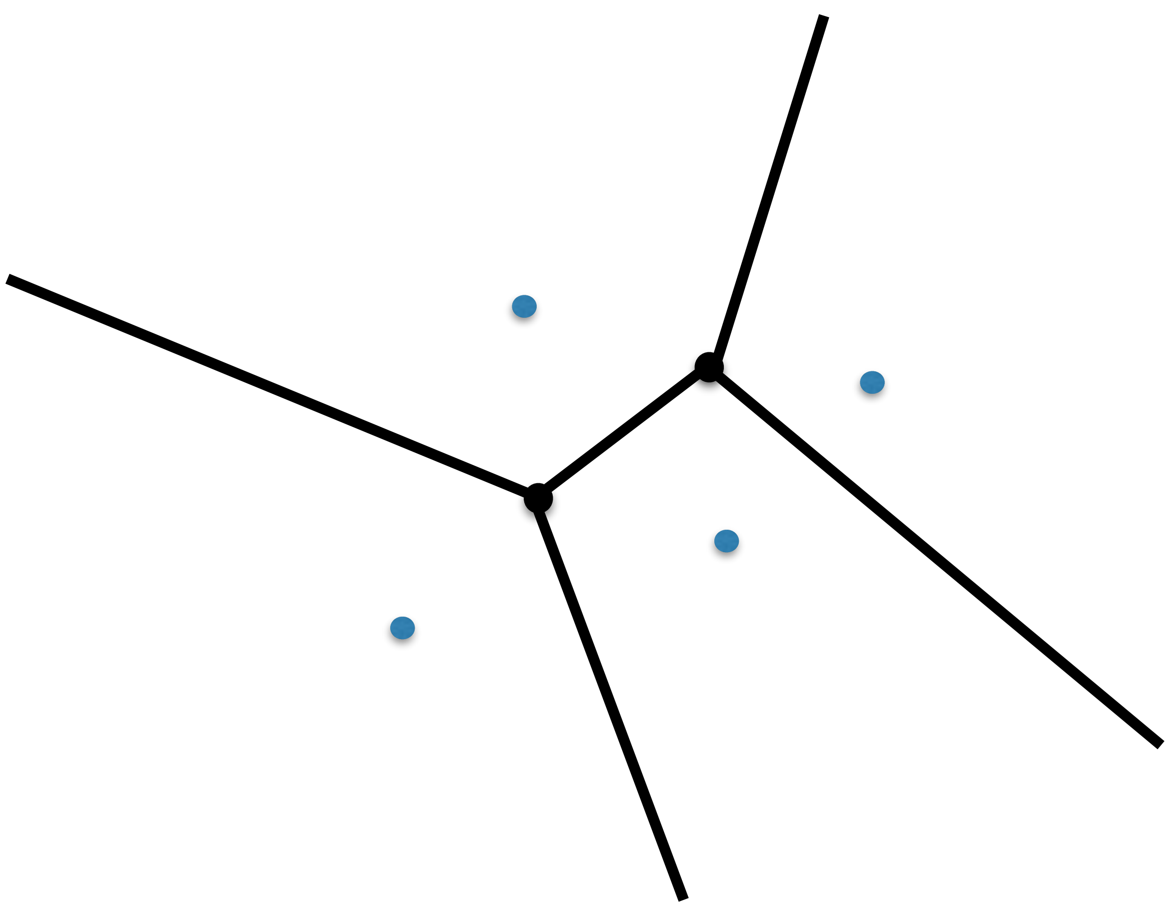
- Voronoi edges
  - The edges of Vor(P) are segments of perpendicular bisectors
  - Each Voronoi edge bounds two Voronoi cells, say Vor(p_i) and Vor(p_j) and must lie on the perpendicular bisector of p_i and p_j
  - Each point on an edge is equidistant from p_i and p_j, and p_i and p_j are its closest sites



Properties of Voronoi Diagram

Let P = {p_1, p_2,…,p_n} set of points in the plane.

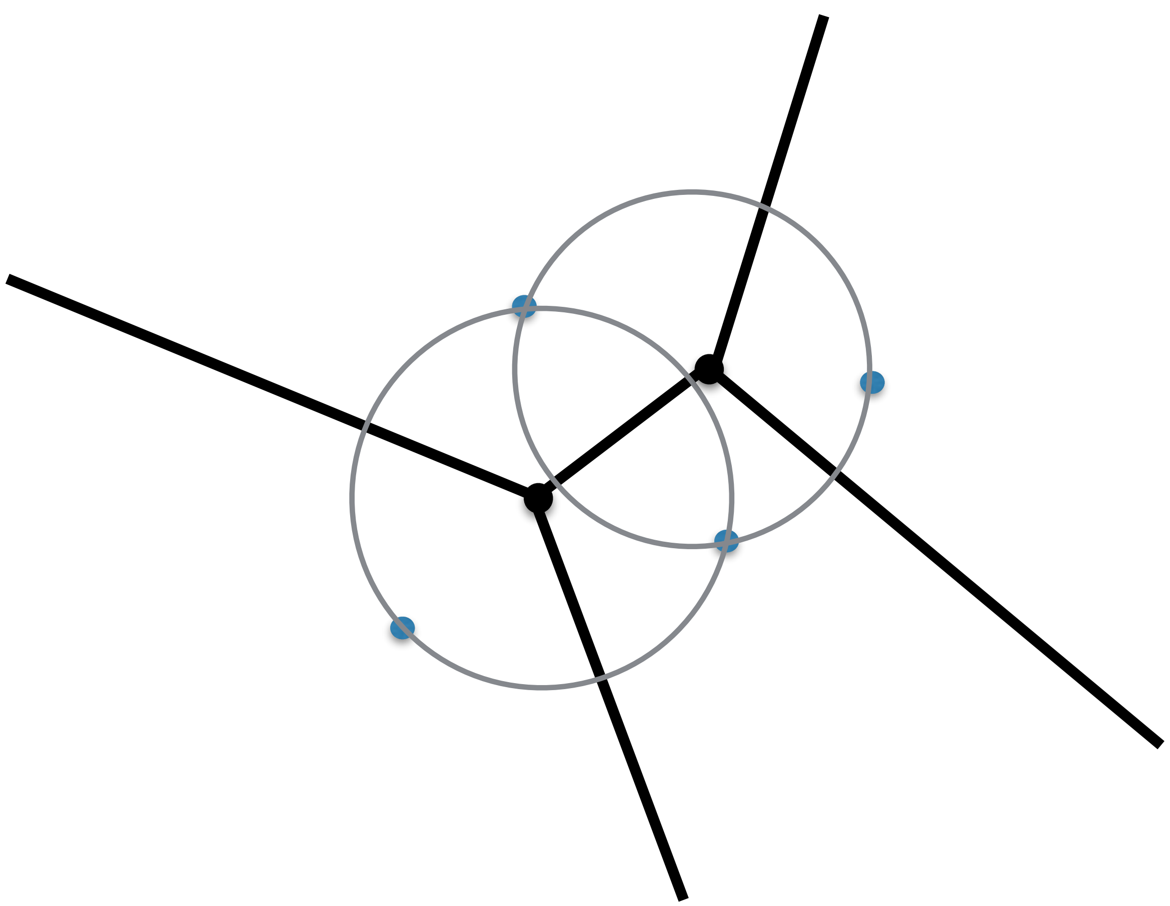
- Voronoi vertices
  - The points where 3 or more Voronoi cells intersect is called a **Voronoi vertex**
  - A Voronoi vertex is equidistant from those sites



Properties of Voronoi Diagram

Let $P = \{p_1, p_2, \ldots, p_n\}$ set of points in the plane.

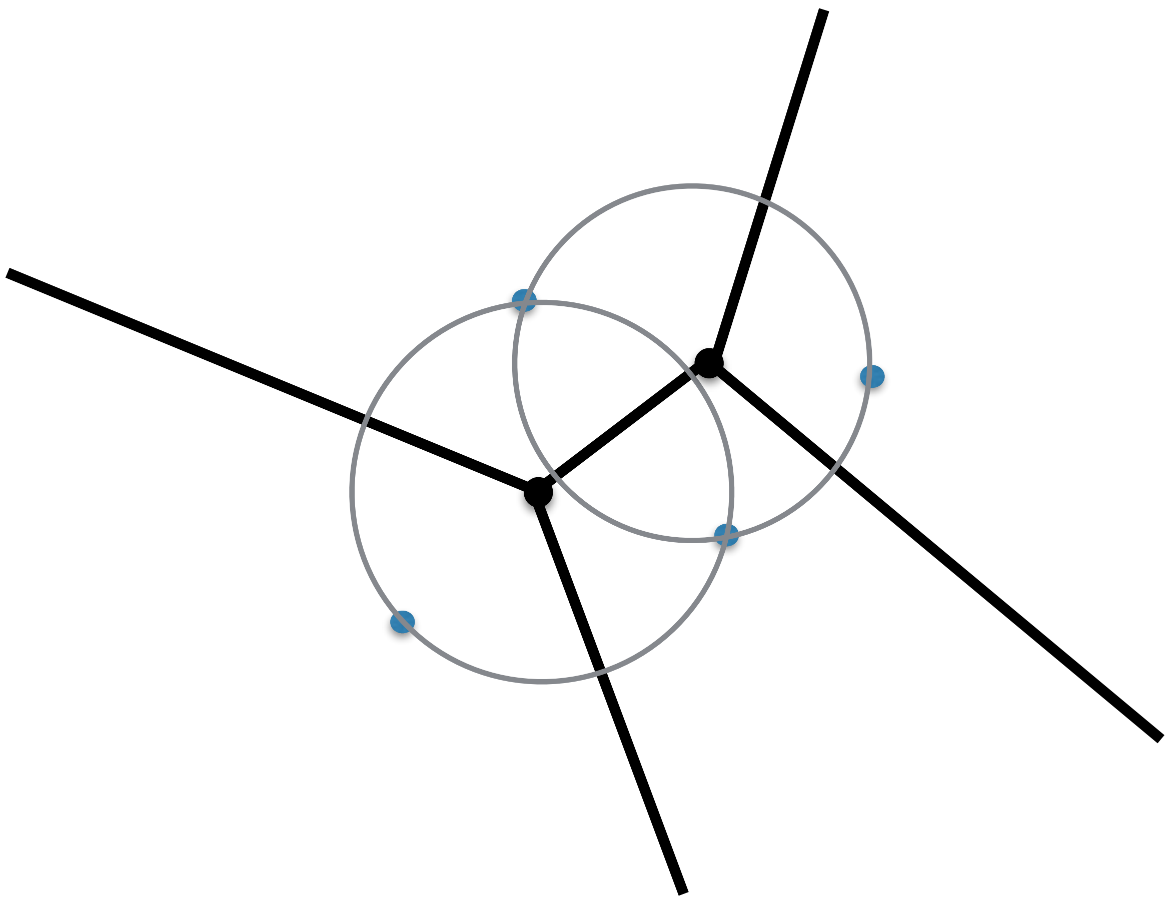
- Voronoi vertices
  - The points where 3 or more Voronoi cells intersect is called a **Voronoi vertex**
  - A Voronoi vertex is equidistant from those sites



Properties of Voronoi Diagram

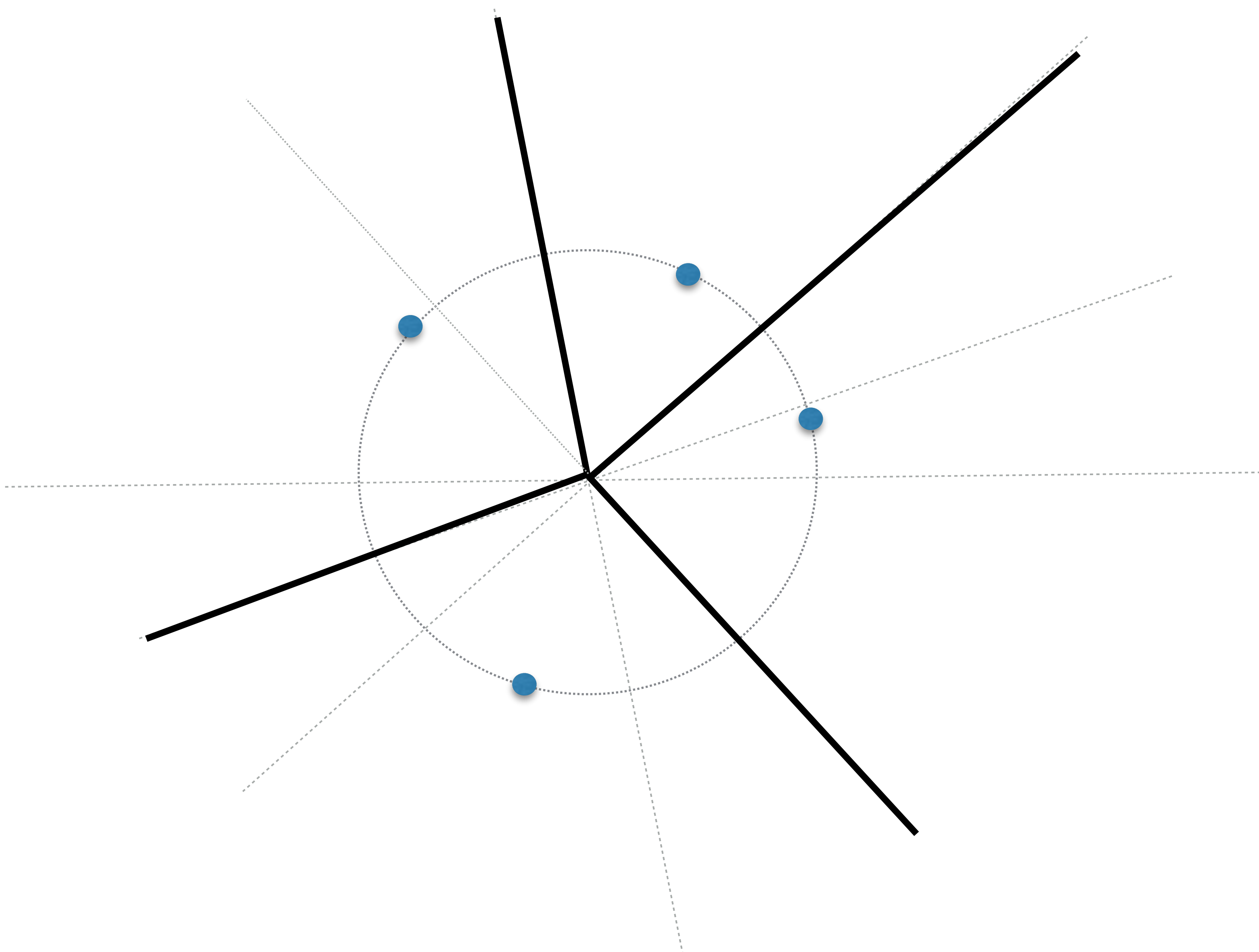
Let $P = \{p_1, p_2, \ldots, p_n\}$ set of points in the plane.

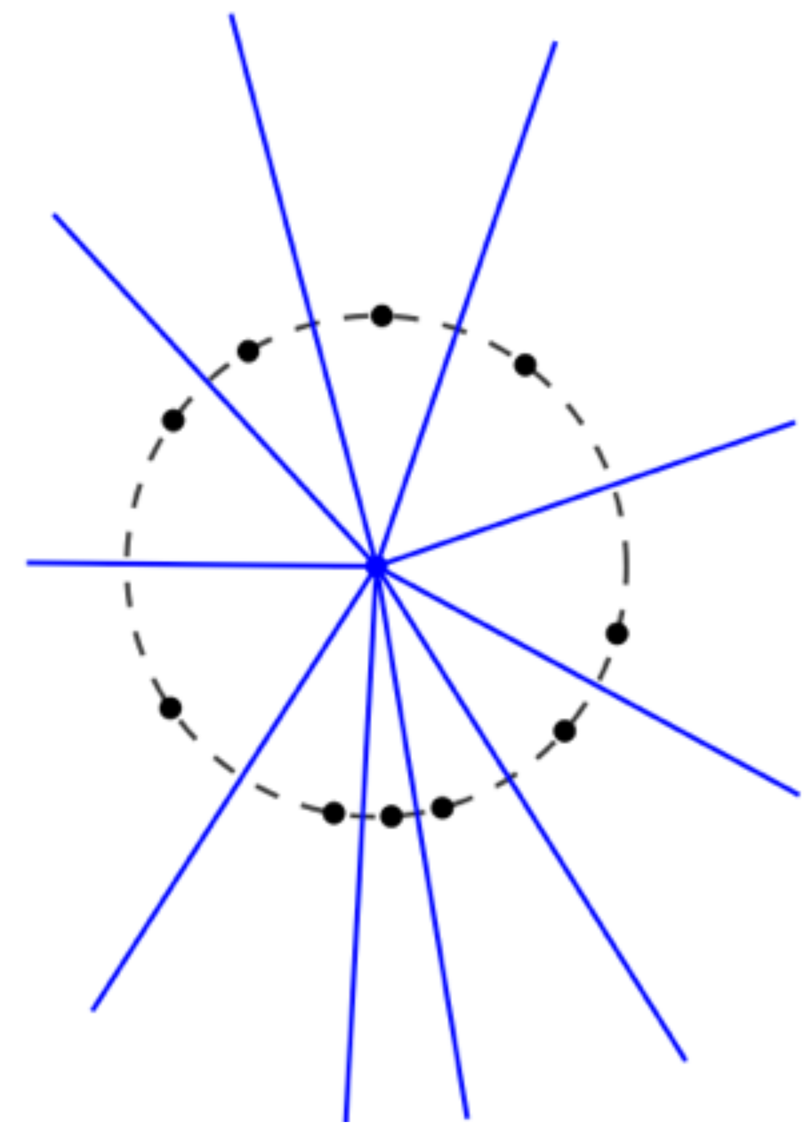
- Voronoi vertices
  - The points where 3 or more Voronoi cells intersect is called a **Voronoi vertex**
  - A Voronoi vertex is equidistant from those sites
  - Can a Voronoi vertex have degree > 3 ? Draw an example.



Degeneracies

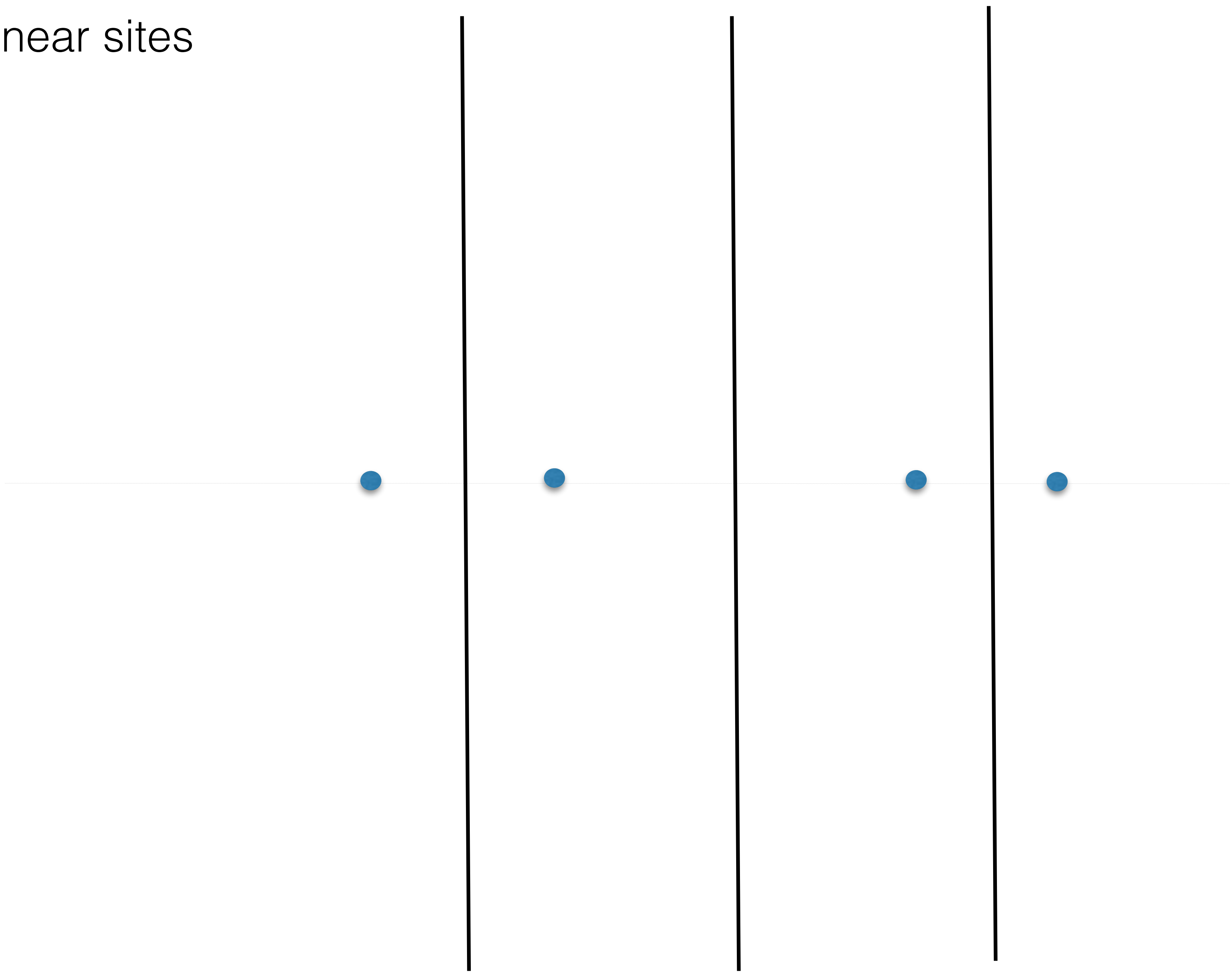
- More than 3 sites lie on the same circle





Degeneracies

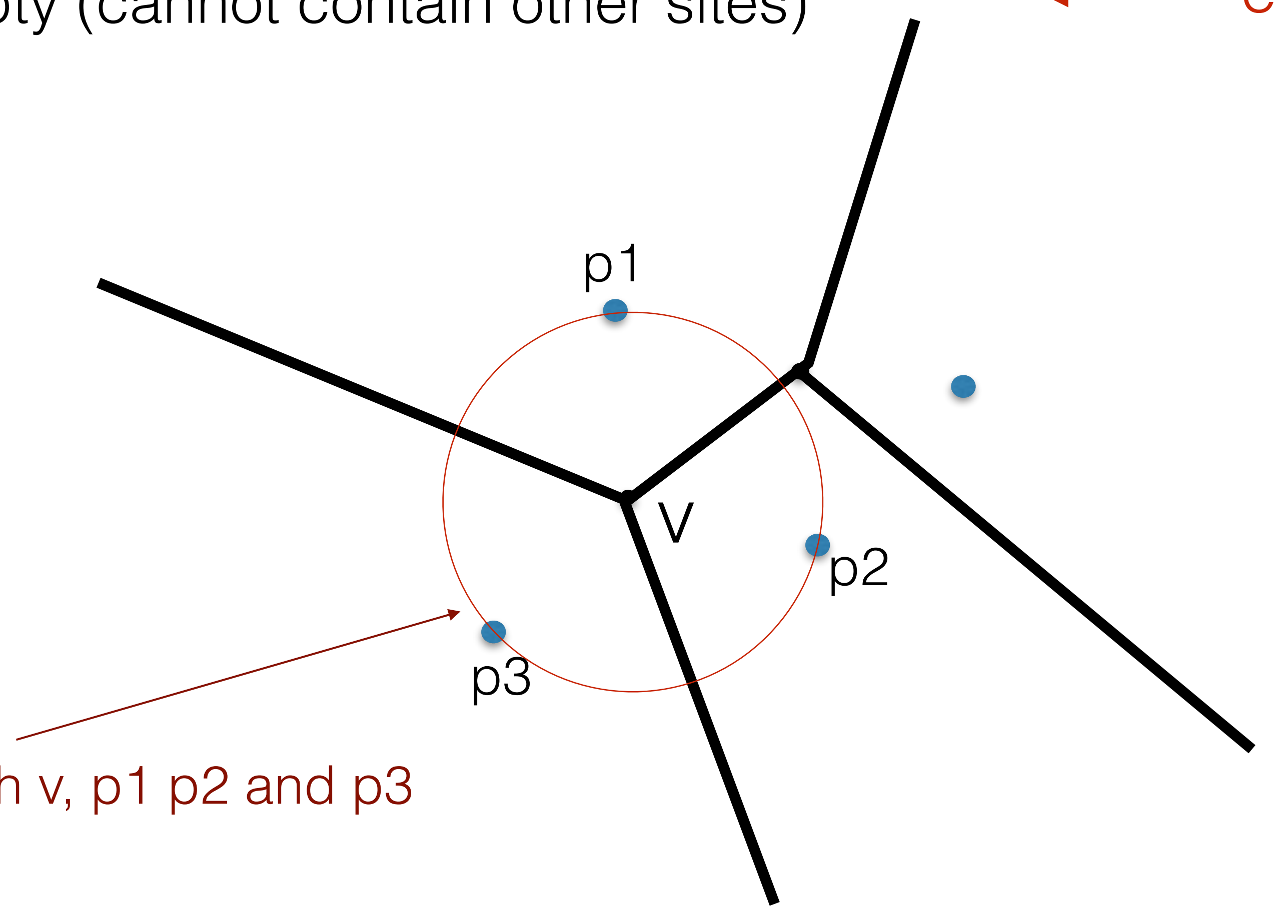
- Collinear sites



Properties of Voronoi Diagram

Let P = {p_1, p_2,…,p_n} set of points in the plane **such that no 4 are co-circular**.

- Any Voronoi vertex v
  - Is the intersection of precisely 3 regions, say p1, p2 and p3
  - v is equidistant from p1, p2 and p3
  - Furthermore, p1, p2 and p3 are its nearest neighbors
  - C(v) is empty (cannot contain other sites) ← empty circle property

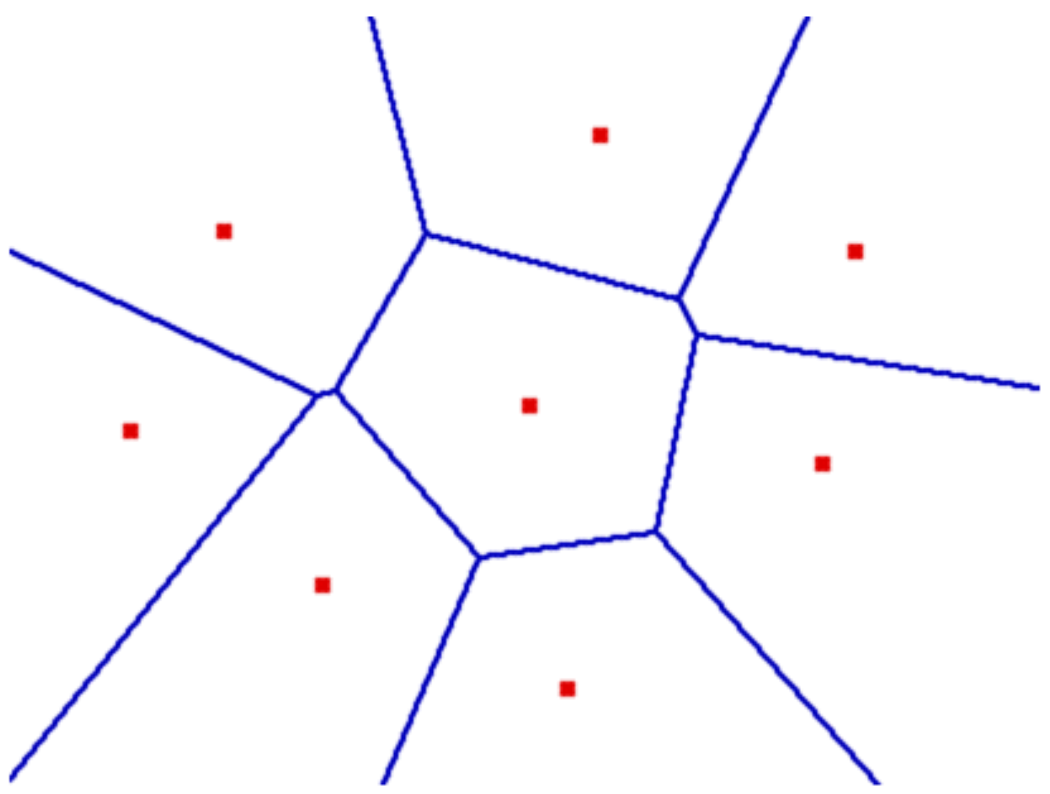


C(v): circle through v, p1 p2 and p3

Properties of Voronoi Diagram

Let $P = \{p_1, p_2, \ldots, p_n\}$ set of points in the plane such that no 4 are co-circular.

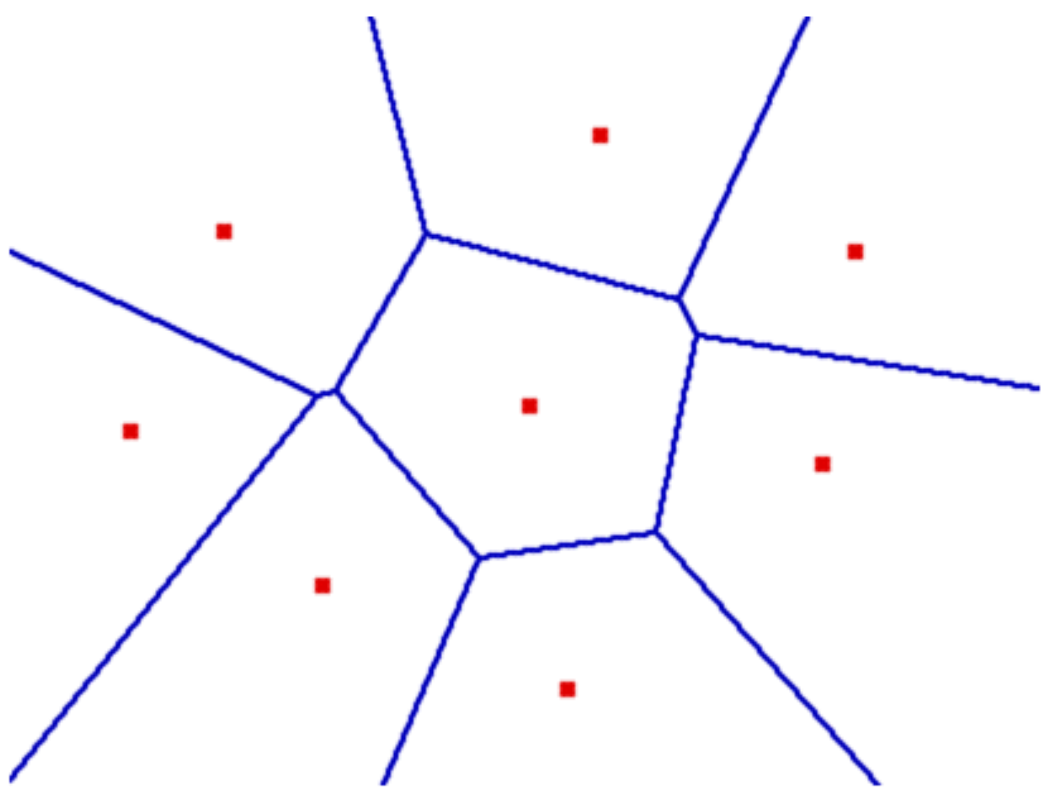
- Voronoi regions (cells) can be bounded or unbounded
- Claim: A point p is on the convex hull of P if and only if Vor(p) is unbounded.



Properties of Voronoi Diagram

Let $P = \{p_1, p_2, \ldots, p_n\}$ set of points in the plane such that no 4 are co-circular.

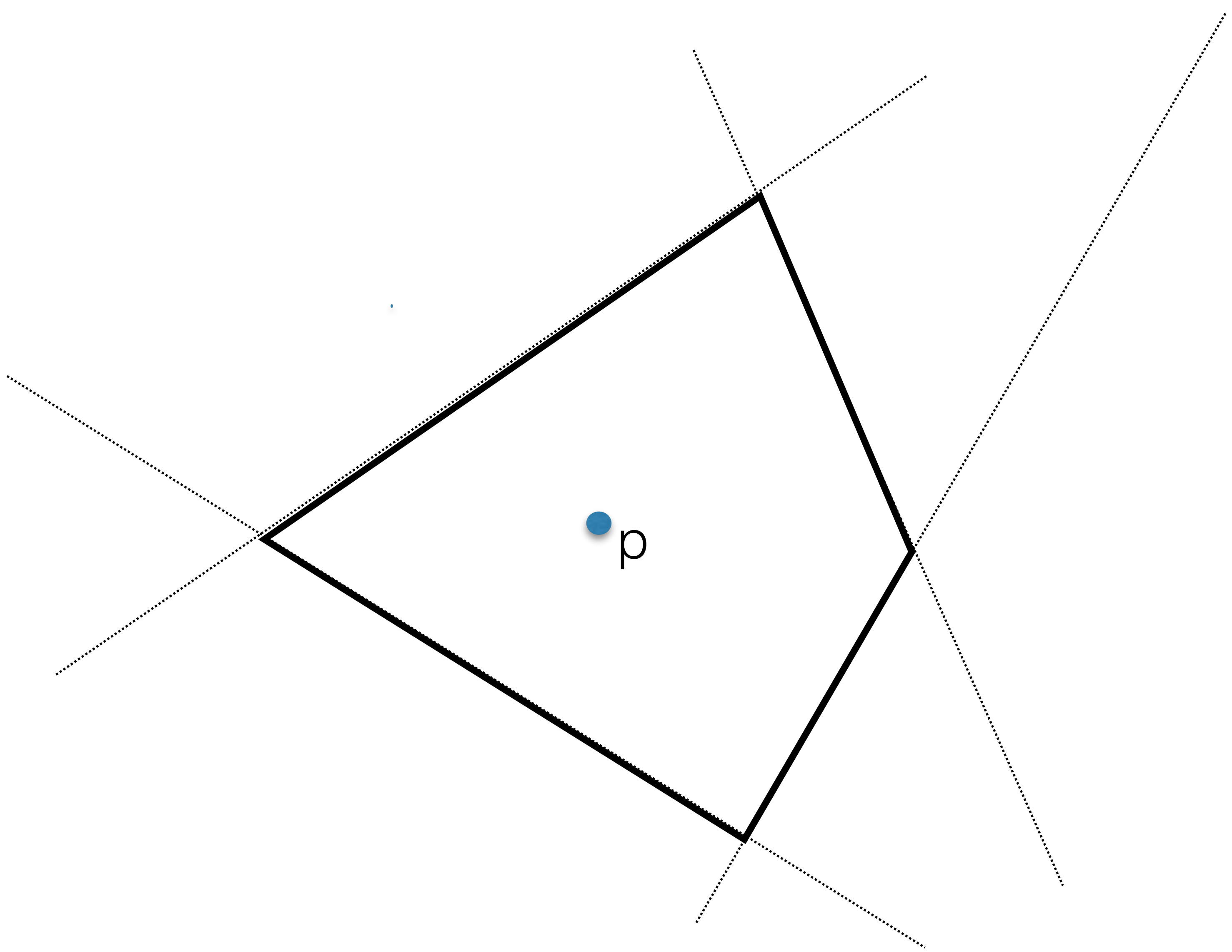
- Voronoi regions (cells) can be bounded or unbounded
- Claim: A point p is on the convex hull of P if and only if Vor(p) is unbounded.



- This means that if we computed Vor(P), we can find CH(P) in linear time.

Properties of Voronoi Diagram

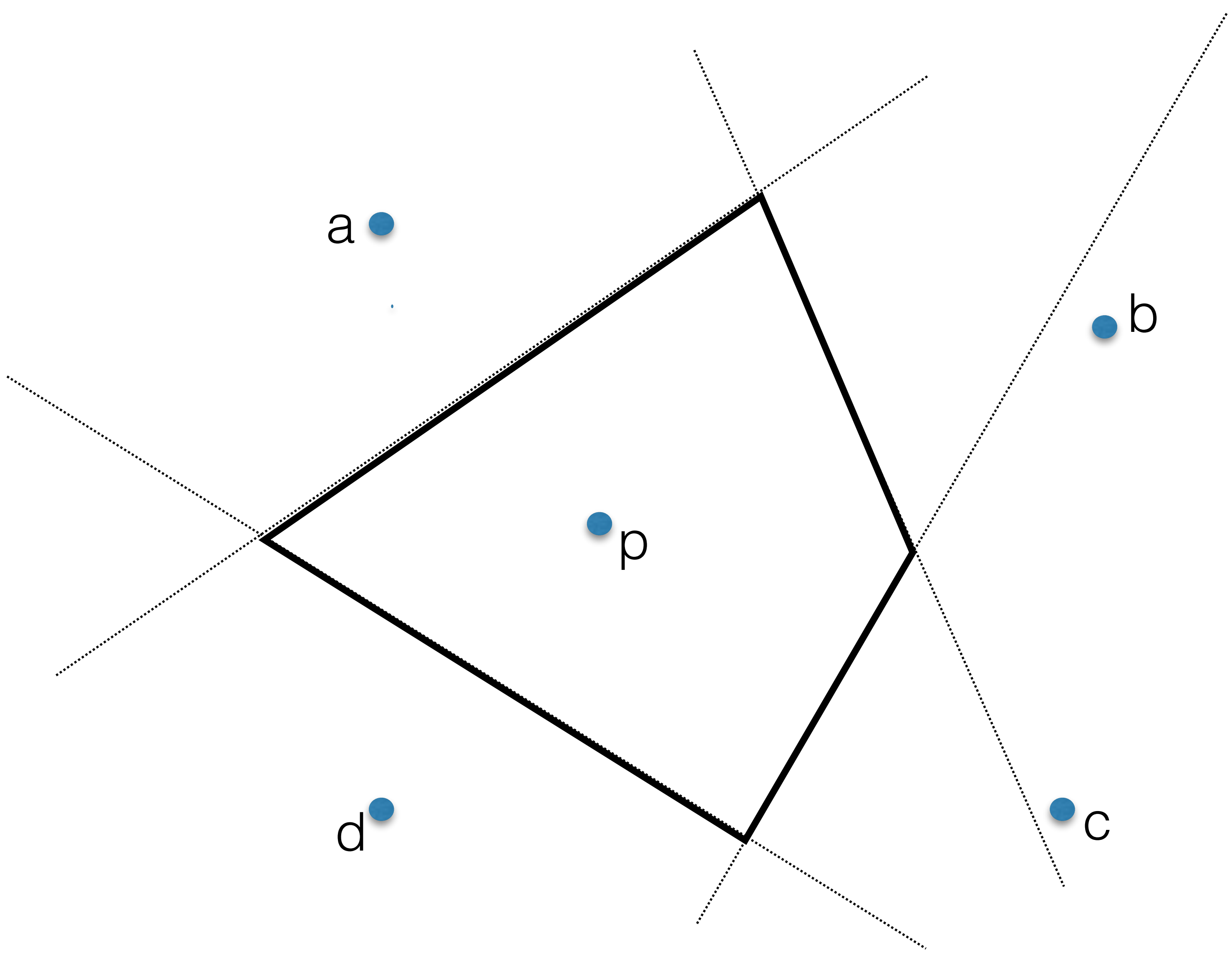
Claim: If Vor(p) is bounded => p inside the CH



Properties of Voronoi Diagram

Claim: If Vor(p) is bounded => p inside the CH

Proof: Consider a point p with Vor(p) a bounded convex polygon. Each edge belongs to a perpendicular bisector. In any direction around p, there is a site beyond the edge. p must be inside polygon abcd => p is inside the CH.



Properties of Voronoi Diagram

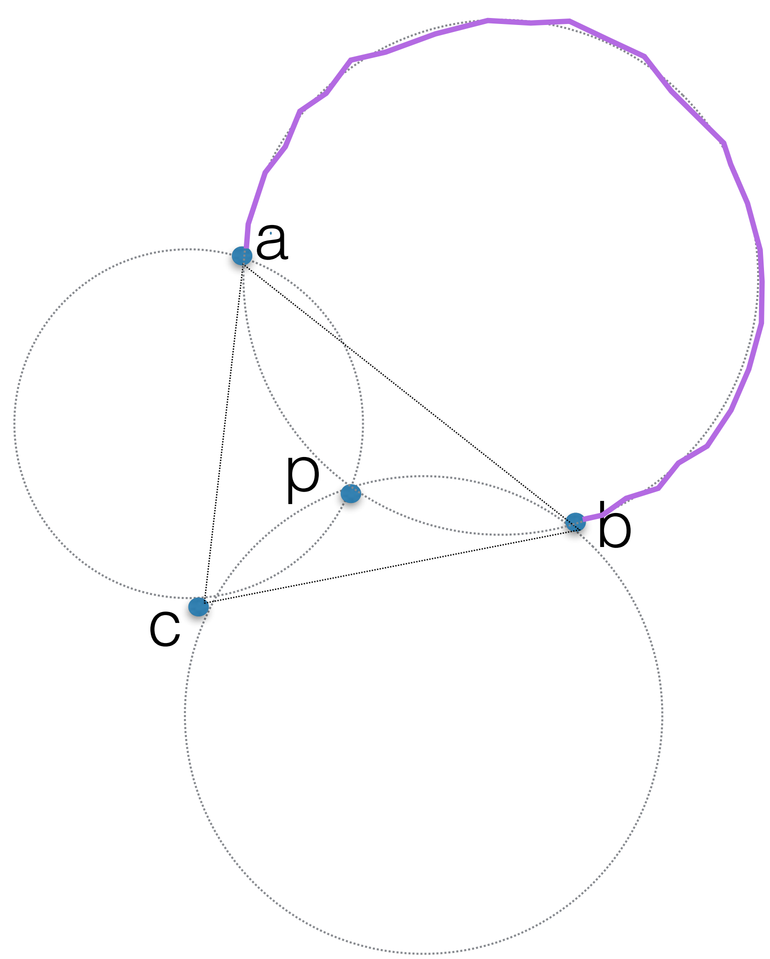
Claim: If p inside the CH => Vor(p) is bounded

Proof:

If p is inside the CH, there must exist a triangle abc containing p. Consider the circles through pab, pac and pbc.

It can be shown that any point outside these circles cannot have p as its closest site.

This means the region of p must be contained within these circles.



Any point on this arc is closer to one of {a,b} than to p

Size of Vor(P)

Let $P = \{p_1, p_2, \ldots, p_n\}$ set of points in the plane such that no 4 are co-circular.

Exercise

- Design a set of points such that the Voronoi cell of one vertex has n-1 edges.

Size of Vor(P)

Let P = {p_1, p_2,…,p_n} set of points in the plane such that no 4 are co-circular.

- The upper bound for a cell in the Voronoi diagram is O(n)
- Therefore, a trivial bound on the size of Vor(P) is $O(n^2)$

Claim: The total size of Vor(P) is O(n).

Proof:

- Vor(P) is a planar graph with n faces.
- Each Voronoi vertex is adjacent to exactly 3 edges, $e = 3v$
- By Euler theorem, $v - e + f = 2$
- Since $f=n$ and $e = 3v$ it follows that the number of Voronoi vertices and edges are O(n) as well.

Computing Voronoi diagrams

- Naive algorithm
    - For each site, compute its cell as the intersection of n-1 bisector halfplanes
    - The intersection of in halfplanes can be found in O(n lg n)
    - This leads to an $O(n^2 \lg n)$ algorithm
- Incremental construction
    - For each point p_i, insert p_i in the Voronoi diagram of previous points
    - The diagram changes only “locally” and insertion can be done in O(n)
    - Overall $O(n^2)$
- Plane sweep
    - Fortune’s algorithm runs in O(n lg n)
    - Simple (in retrospect) and elegant
- Randomized incremental construction
    - Runs in average in O(n lg n)
    - Good (best?) in practice

Applications

- Vor(P) stores everything there is to know about proximity
- Many applications in many disciplines
  - Proximity problems
  - Facility location
  - Interpolation
    - natural neighbor interpolation based on Voronoi region of p
  - Morphology
  - Art
  - Personal spaces
  - …

from Wikipedia

Applications

- In biology, Voronoi diagrams are used to model a number of different biological structures, including cells[13] and bone microarchitecture.[14] Indeed, Voronoi tessellations work as a geometrical tool to understand the physical constraints that drive the organization of biological tissues.

- In hydrology, Voronoi diagrams are used to calculate the rainfall of an area, based on a series of point measurements. In this usage, they are generally referred to as Thiessen polygons.

- In ecology, Voronoi diagrams are used to study the growth patterns of forests and forest canopies, and may also be helpful in developing predictive models for forest fires.

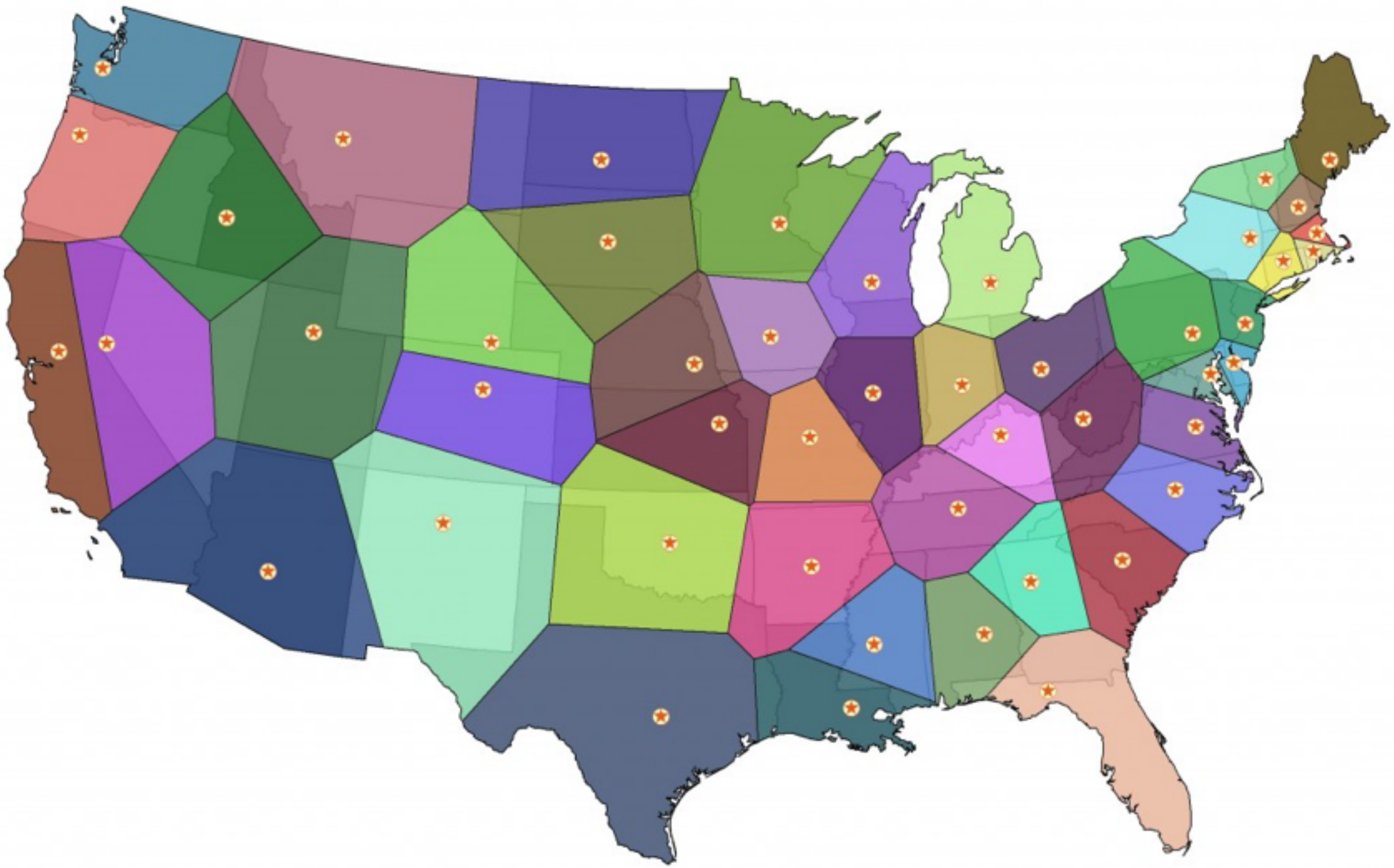
- In computational chemistry, Voronoi cells defined by the positions of the nuclei in a molecule are used to compute atomic charges. This is done using the Voronoi deformation density method.

- In astrophysics, Voronoi diagrams are used to generate adaptative smoothing zones on images, adding signal fluxes on each one. The main objective for these procedures is to maintain a relatively constant signal-to-noise ratio on all the image.

- In computational fluid dynamics, the Voronoi tessellation of a set of points can be used to define the computational domains used in finite volume methods, e.g. as in the moving-mesh cosmology code AREPO.

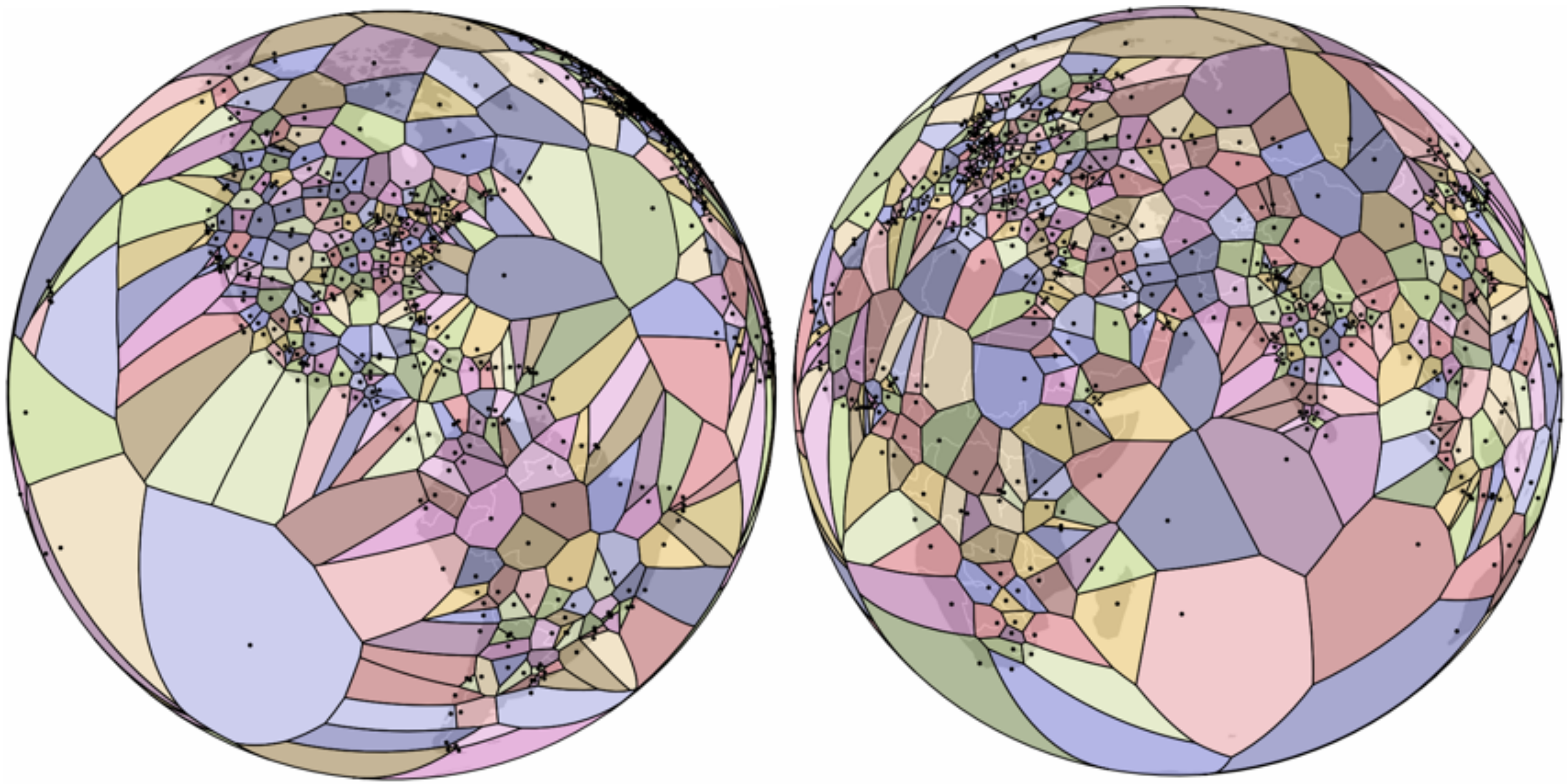
from Wikipedia

- In networking, Voronoi diagrams can be used in derivations of the capacity of a wireless network.
- In computer graphics, Voronoi diagrams are used to calculate 3D shattering / fracturing geometry patterns. It is also used to procedurally generate organic or lava-looking textures.
- In autonomous robot navigation, Voronoi diagrams are used to find clear routes. If the points are obstacles, then the edges of the graph will be the routes furthest from obstacles (and theoretically any collisions).
- In machine learning, Voronoi diagrams are used to do 1-NN classifications.
- In user interface development, Voronoi patterns can be used to compute the best hover state for a given point.
- In epidemiology, Voronoi diagrams can be used to correlate sources of infections in epidemics. One of the early applications of Voronoi diagrams was implemented by John Snow to study the 1854 Broad Street cholera outbreak in Soho, England. He showed the correlation between residential areas on the map of Central London whose residents had been using a specific water pump, and the areas with most deaths due to the outbreak.

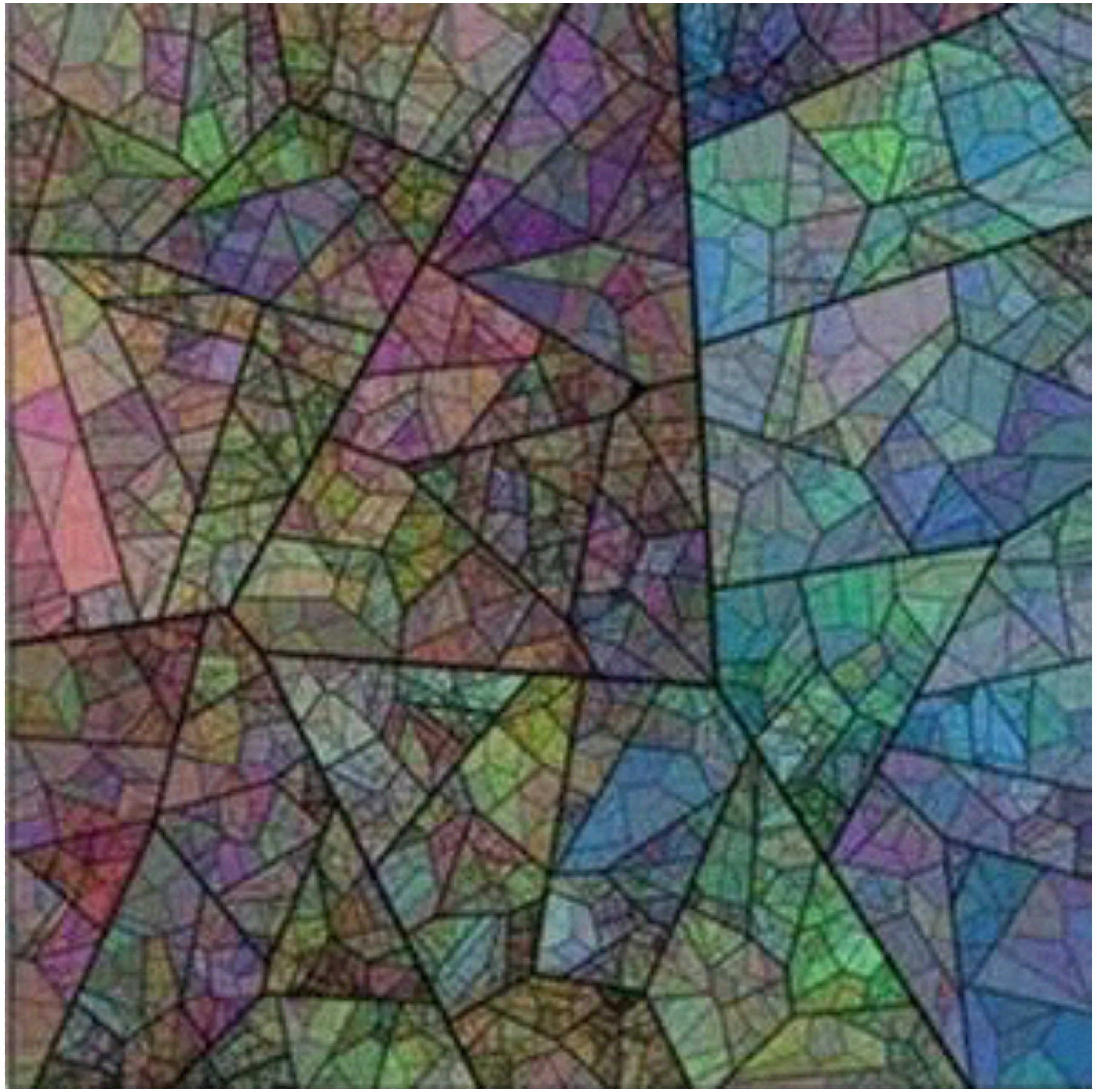


http://2.bp.blogspot.com/_1rwH30ysLko/TNbLbADi3YI/
AAAAAAAACIQ/ObFgwU-CPkY/s1600/
ToddMashup-1024x655.jpg

Closest international Airport

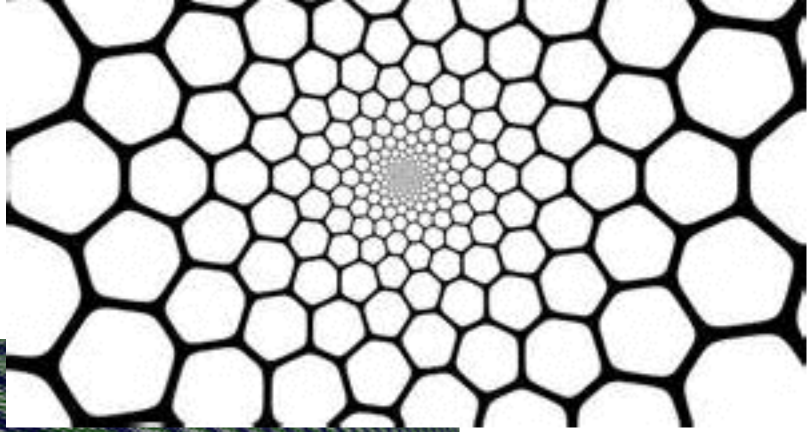
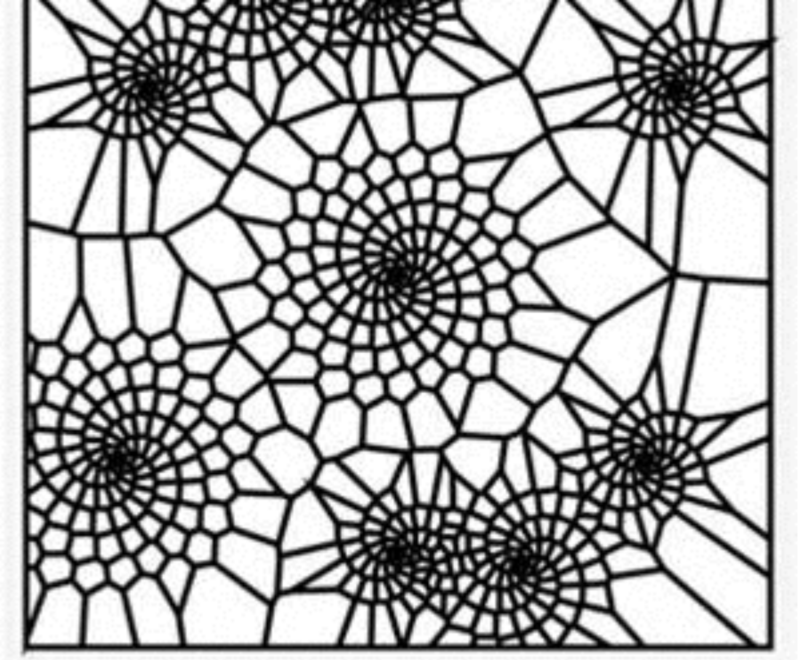
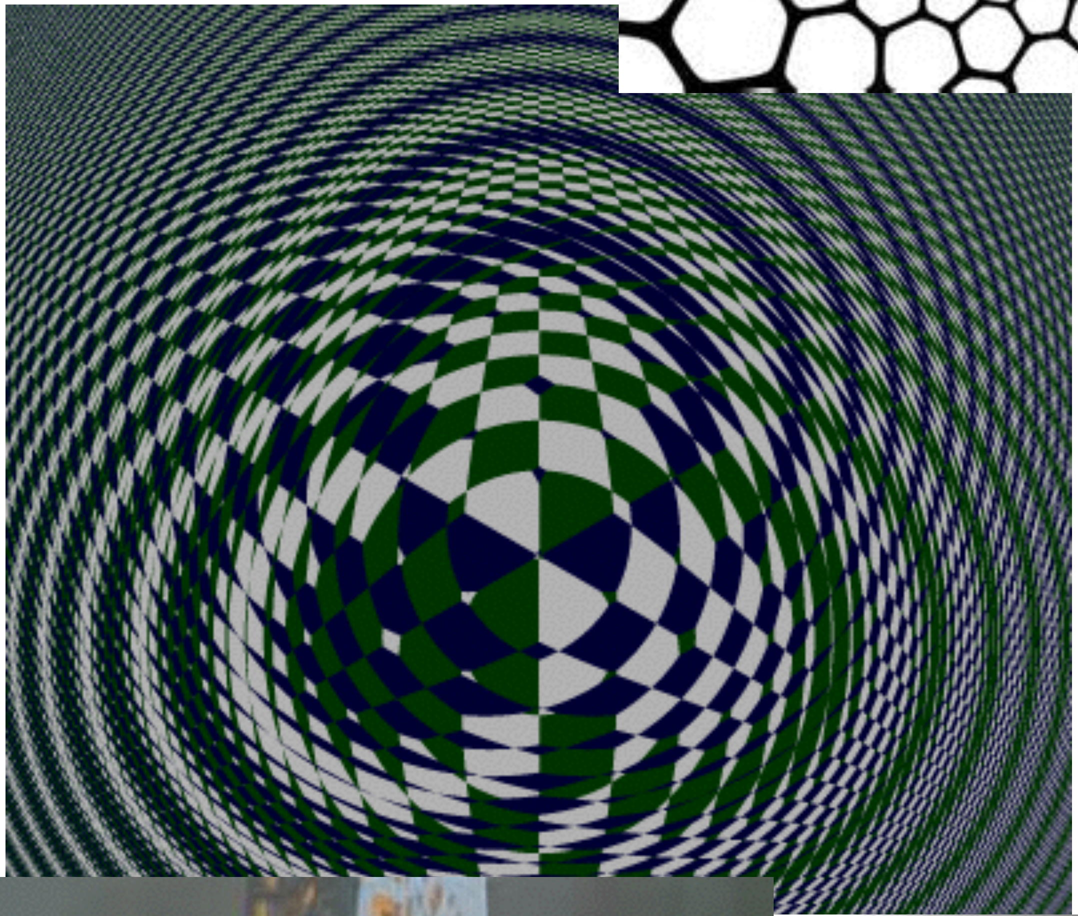
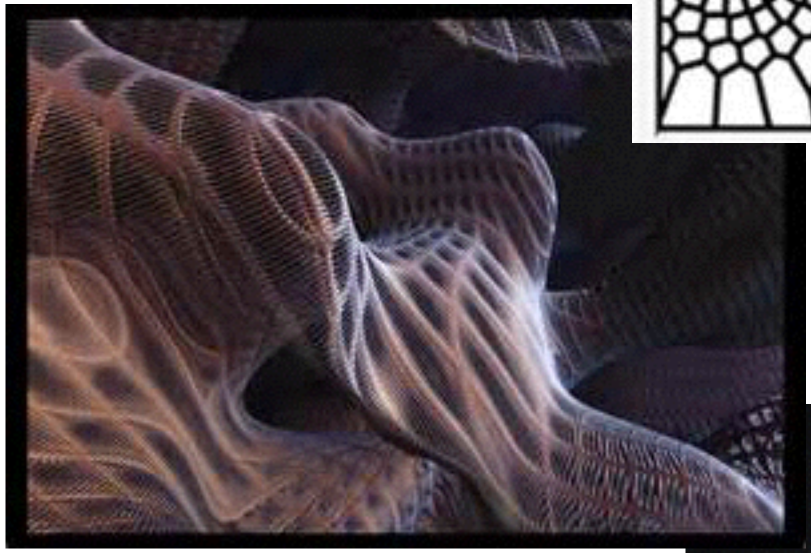
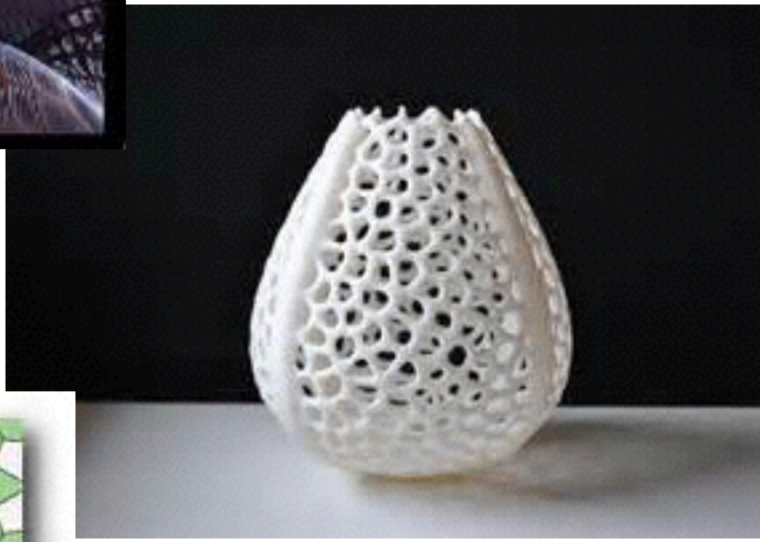


Voronoi art

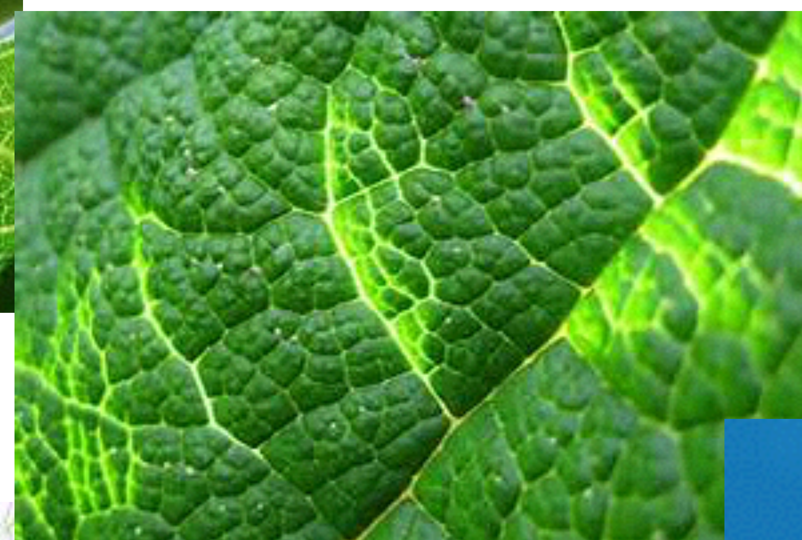


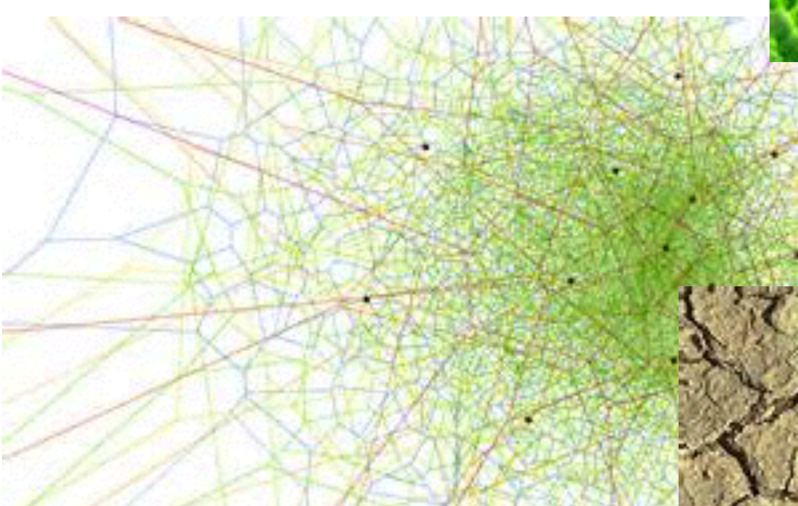
http://www.wblut.com/2008/04/01/voronoi-fractal/

Voronoi art

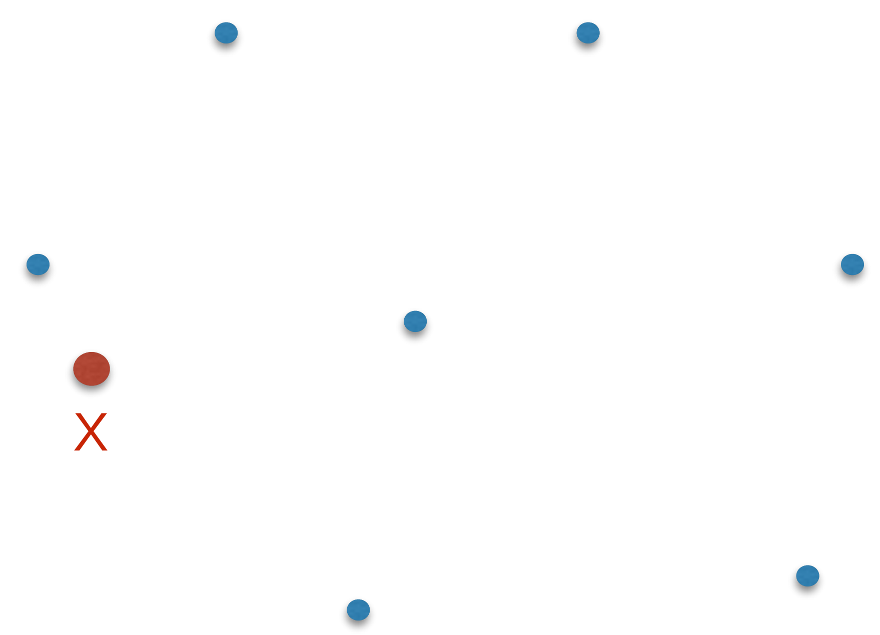

Voronoi in nature



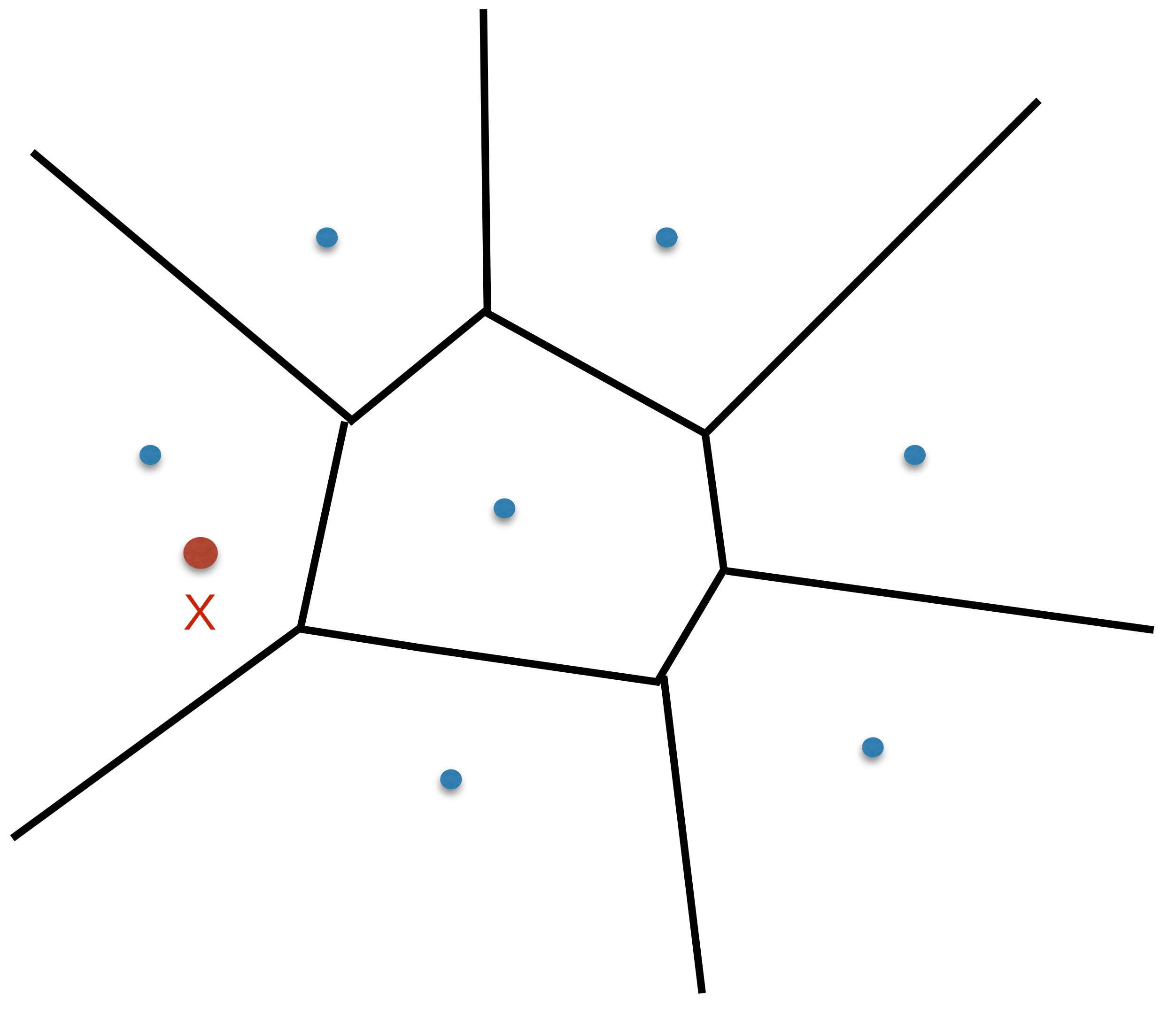
Nearest Neighbor

- Given a set of sites, want to answer **nearest neighbor** queries: Given point x in the plane, find its nearest site.



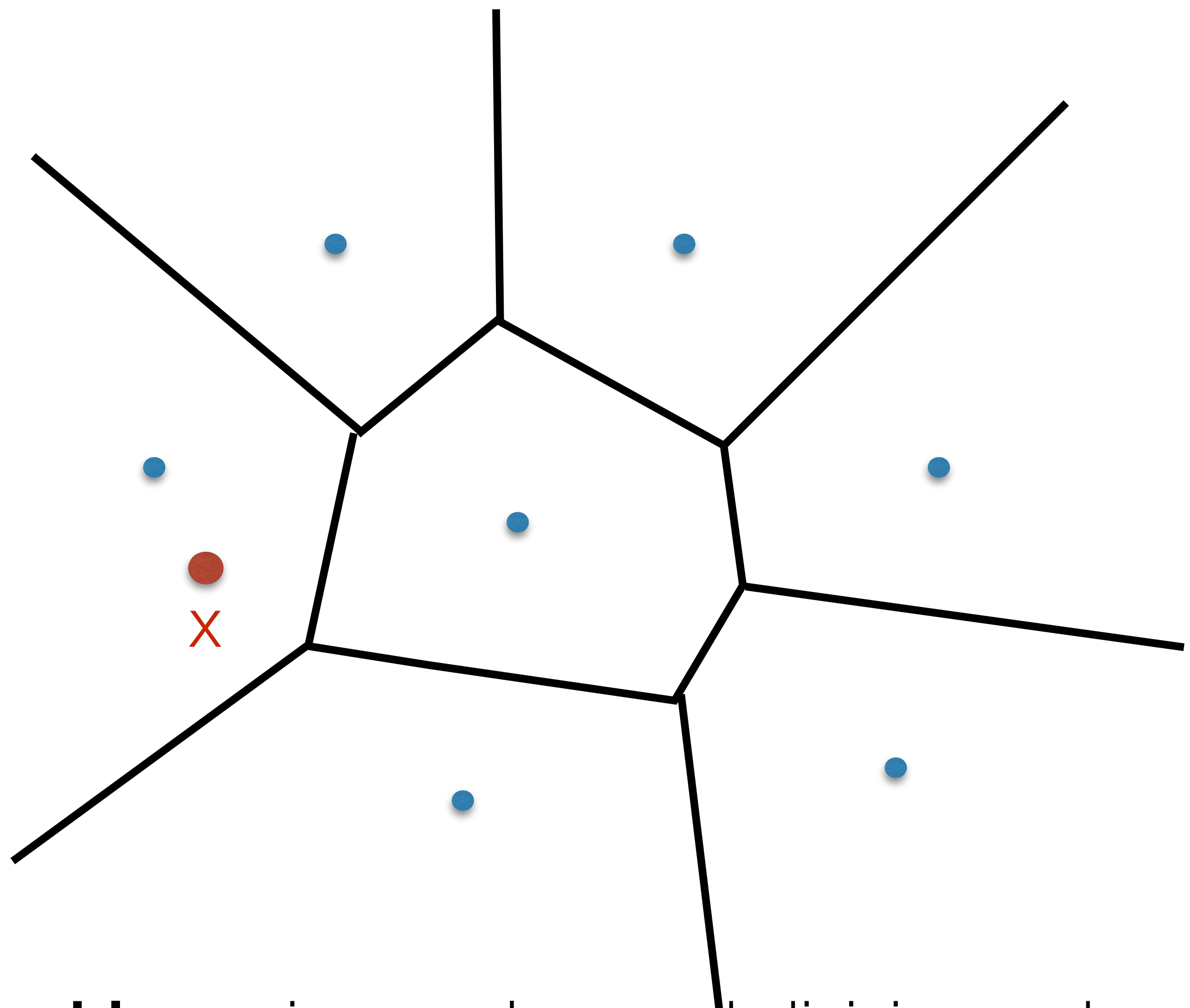
Nearest Neighbor

- Given a set of sites, want to answer **nearest neighbor** queries: Given point x in the plane, find its nearest site.



Nearest Neighbor

- Boils down to solving the point location problem in Vor(P)

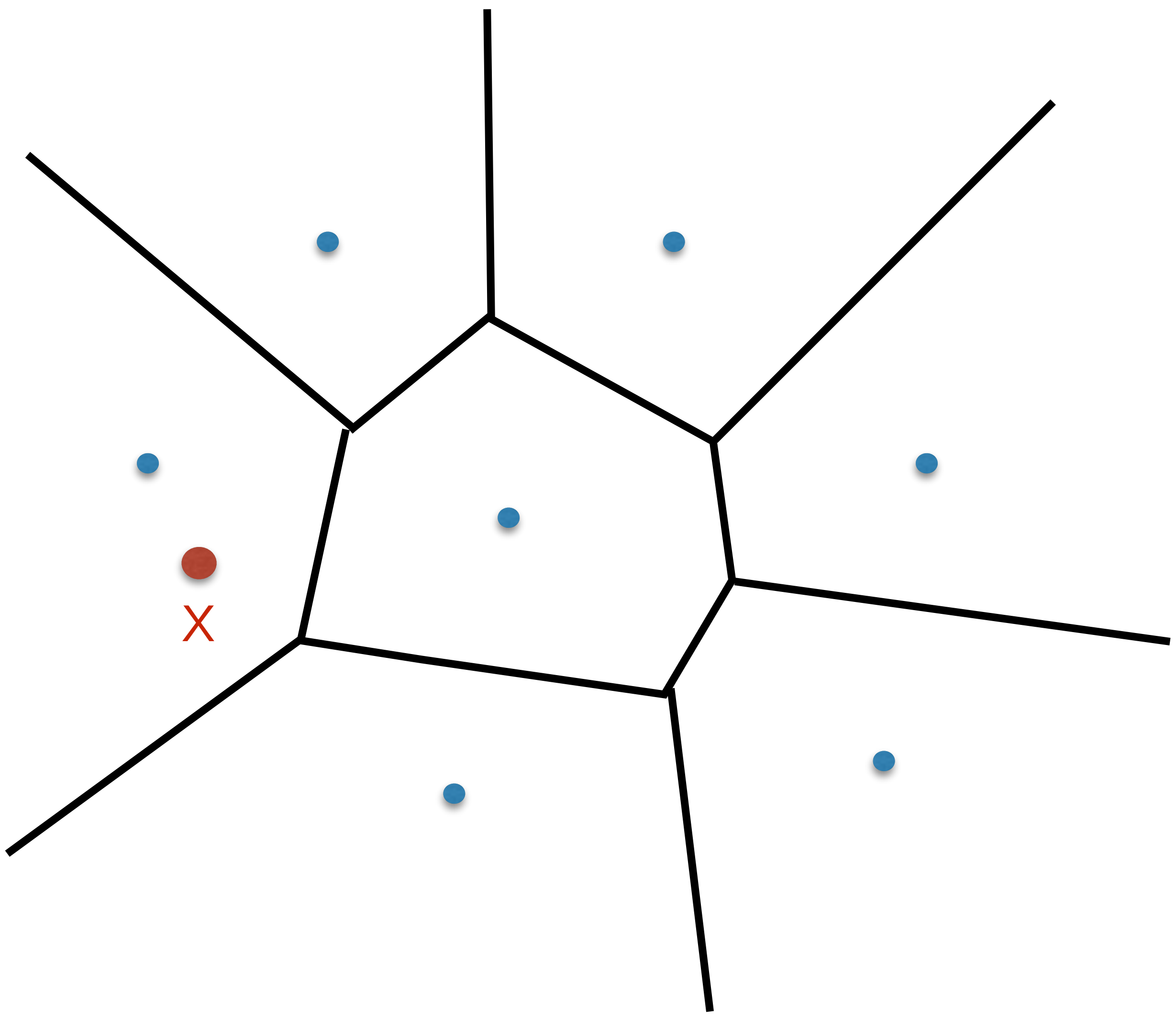


Point location problem: given a planar subdivision and an arbitrary point p, find the region that contains p.

It is known how to pre-process a subdivision into a data structure that can answer point location queries in O(lg n) time.

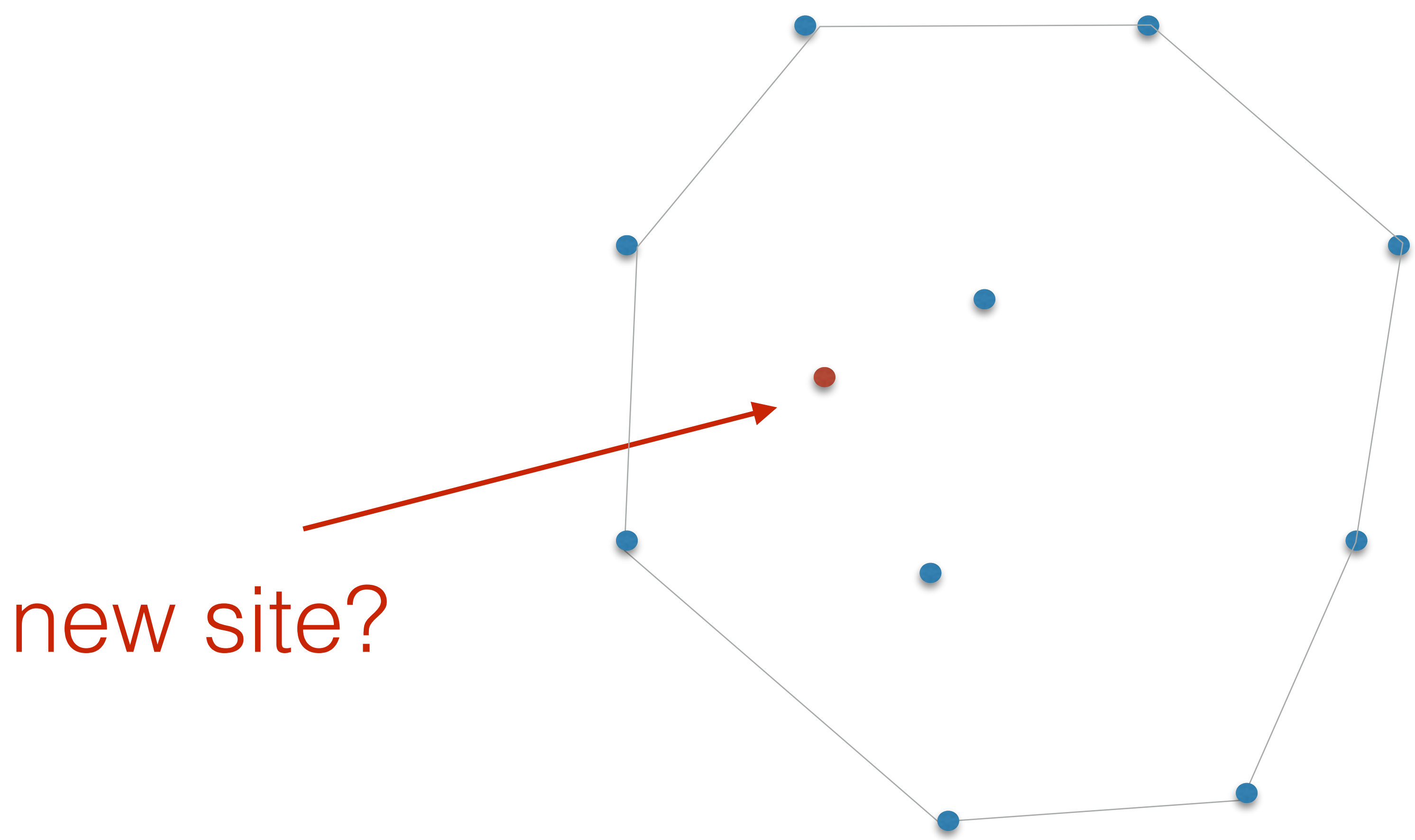
Nearest Neighbor

If Vor(P) is given, nearest neighbor queries can be performed in O(lg n) time with O(n) space and O(n lg n) pre-processing time.



Facility location

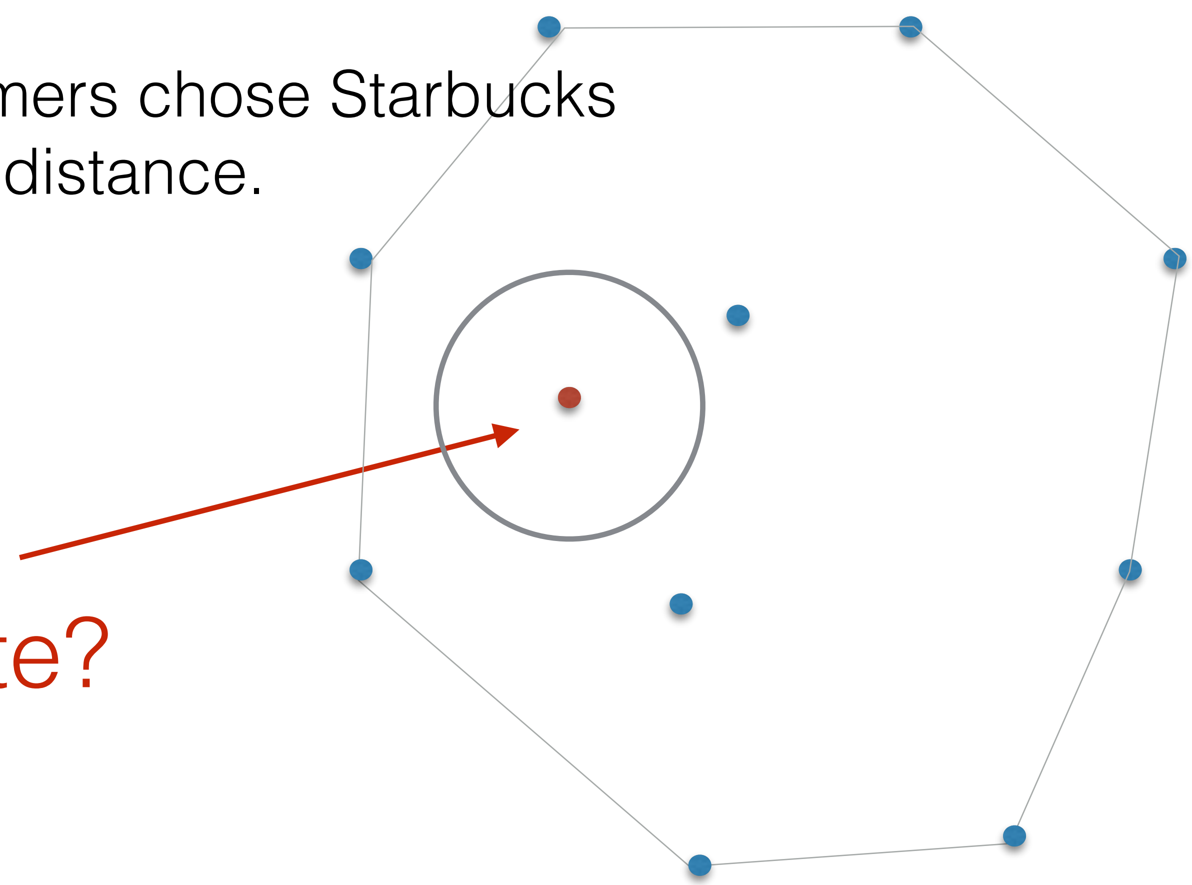
We want to open a new Starbucks. Where should it be placed?



Let’s assume the new site must be inside CH.

We want to open a new Starbucks. Where should it be placed?

Assume customers chose Starbucks
based on their distance.

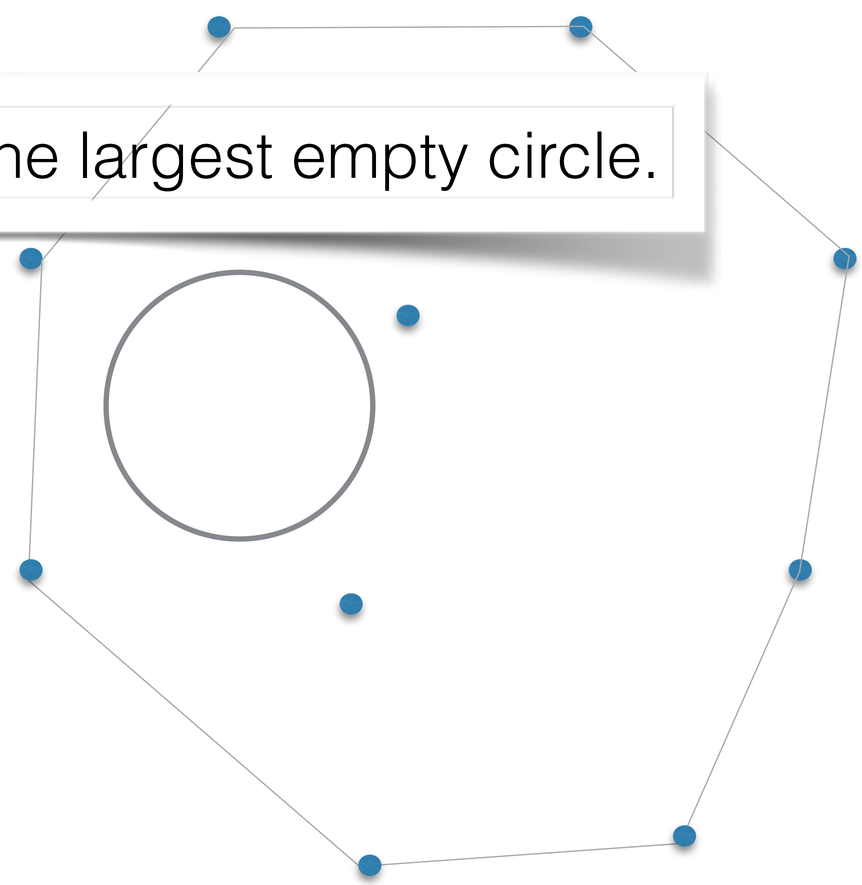


new site?

Let’s assume the new site must be inside CH.

We want to open a new Starbucks. Where should it be placed?

Place it at the center of the largest empty circle.

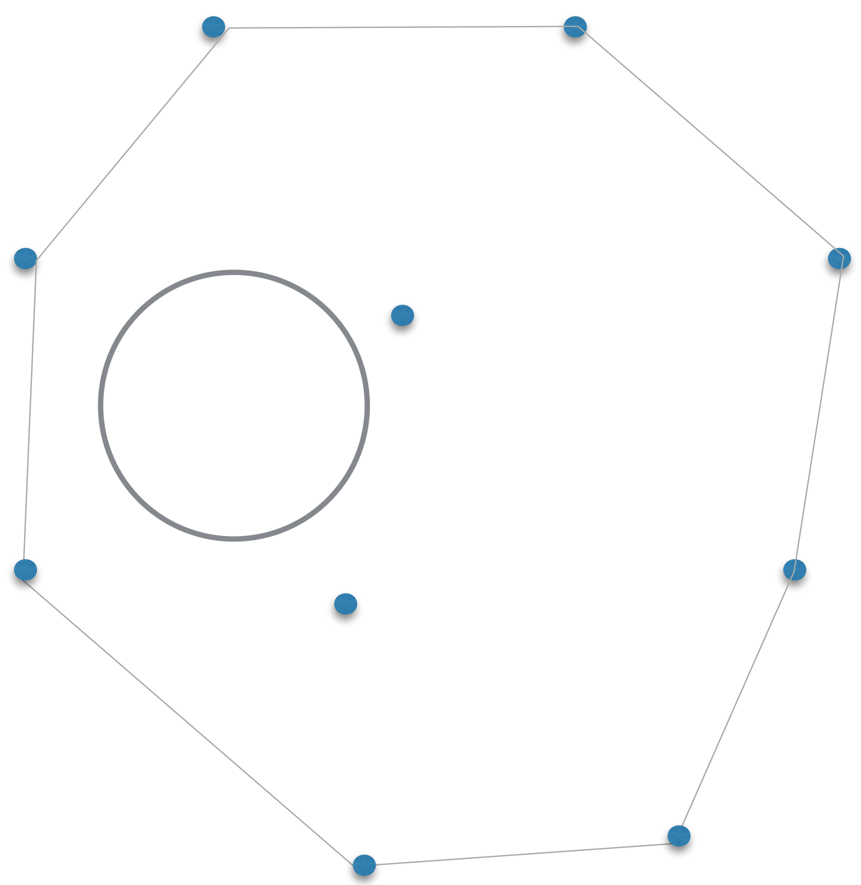


Let’s assume the new site must be inside CH.

Largest empty circle

Given a set P of points, find largest empty circle whose center is strictly inside the hull of P.

Claim: its center must be coincident with a Voronoi vertex.



Proof: Let p be a point in the plane, and let f(p) denote the radius of the largest empty circle centered at p. Imagine how f(p) changes as we move p. We want a point p that achieves max. How to move p to increase f(p)?

Largest empty circle

Given a set P of points, find largest empty circle whose center is strictly inside the hull of P.

Claim: its center must be coincident with a Voronoi vertex.

- Algorithm
  - compute the Voronoi diagram of P
  - For each Voronoi vertex, find the distance to its 3 sites
  - pick the largest

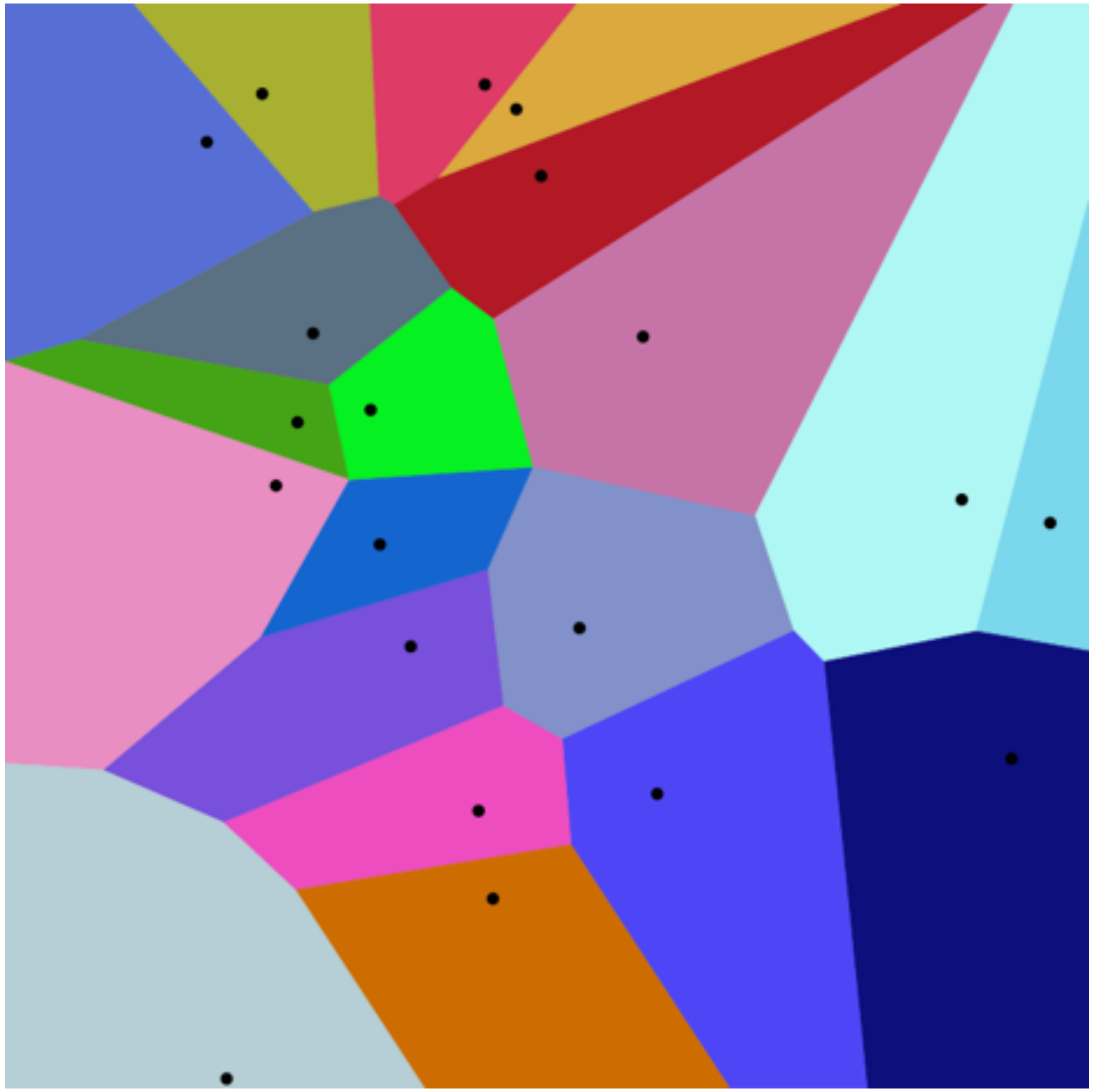
- Time: construct Vor(P) + O(n)

Extensions of Voronoi diagrams

- Vor(P) divides the space according to which site is closest, using Euclidian distance

- Possible extensions
    - higher-order Voronoi diagrams (look at closest k neighbors)
        - Order 2 Voronoi diagrams
            - for any two sites p and q in P, the cell(p,q) is the set of points in the plane whose nearest neighbors are p and q.
        - Farthest-point Voronoi diagram
            - cell(p): all points in the plane for which p is the furthest site
    - use other distance functions
    - d-dimensions

Pictures from Wikipedia



Euclidian distance

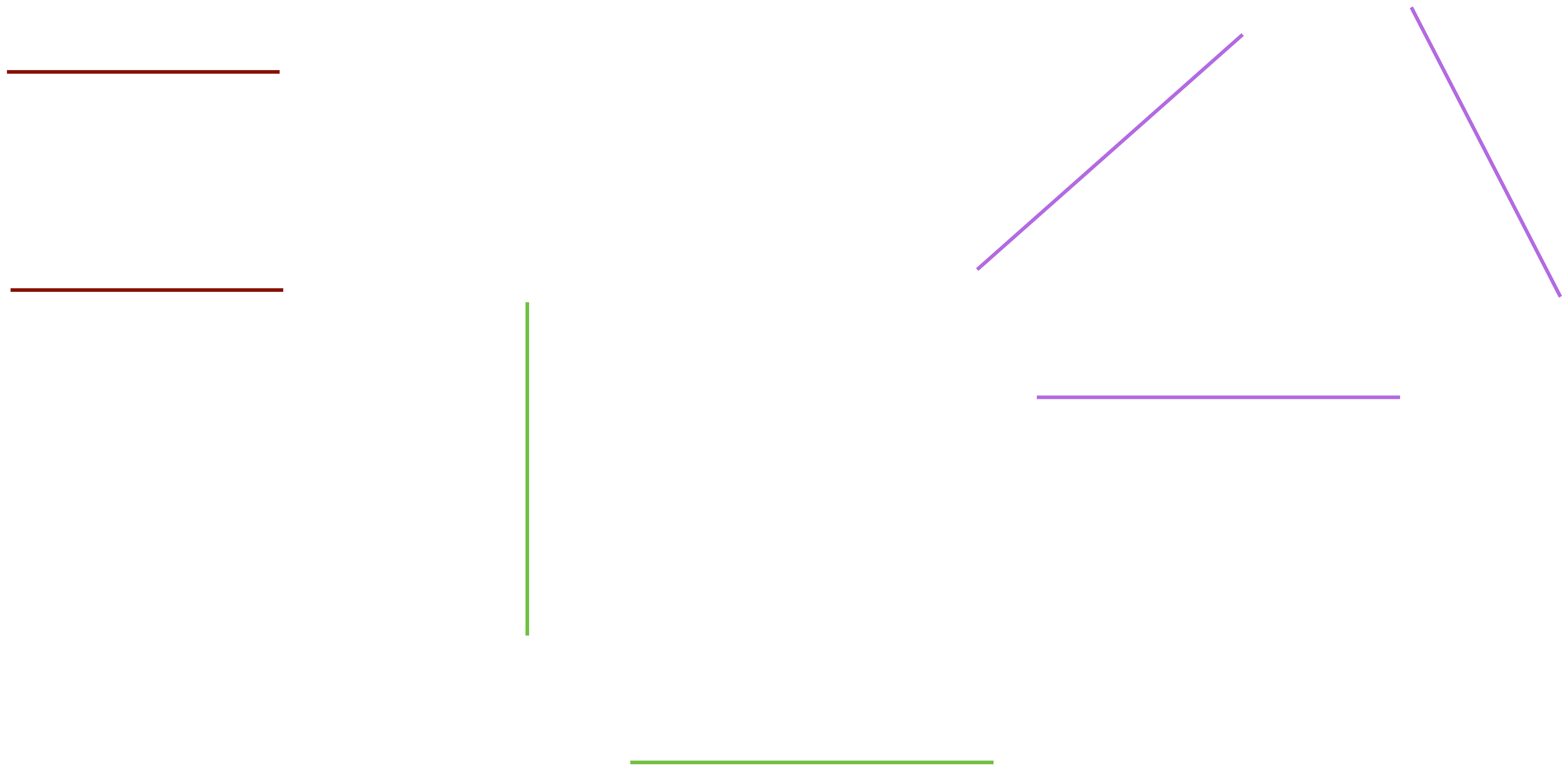
Manhattan distance

Extensions of Voronoi diagrams

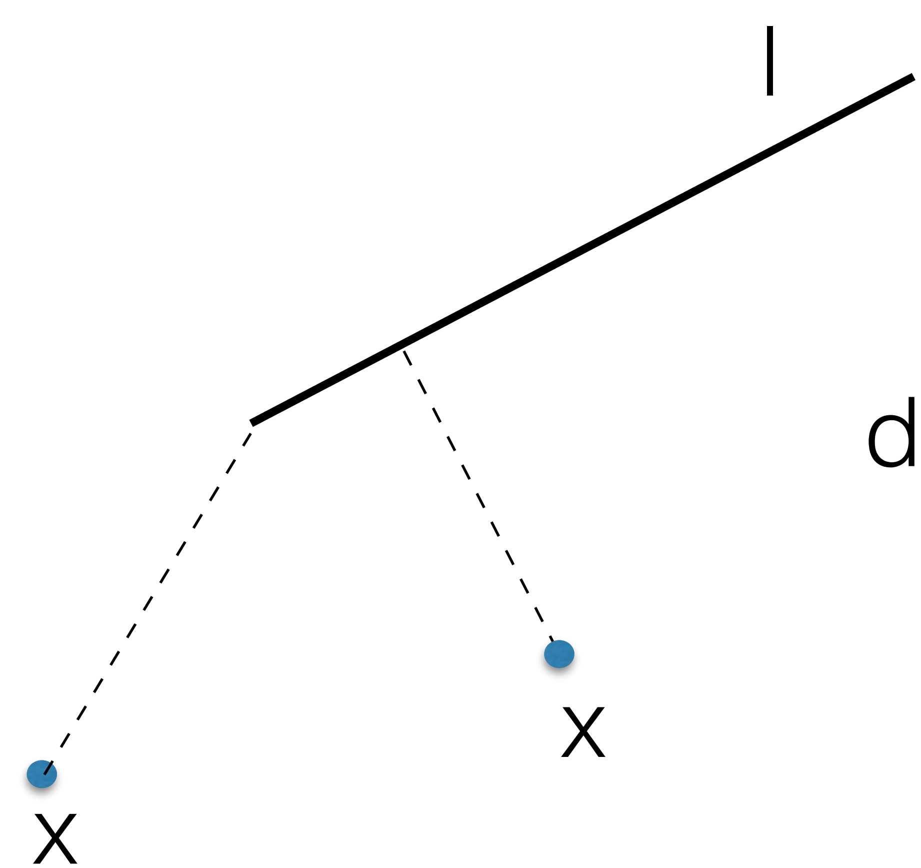
- Voronoi diagram of segments
- Voronoi diagram of polygons
- Medial axis
- 3D
- ..

Voronoi diagram of a set of segments

Given set of segments, partition the plane according to which segment is closest.



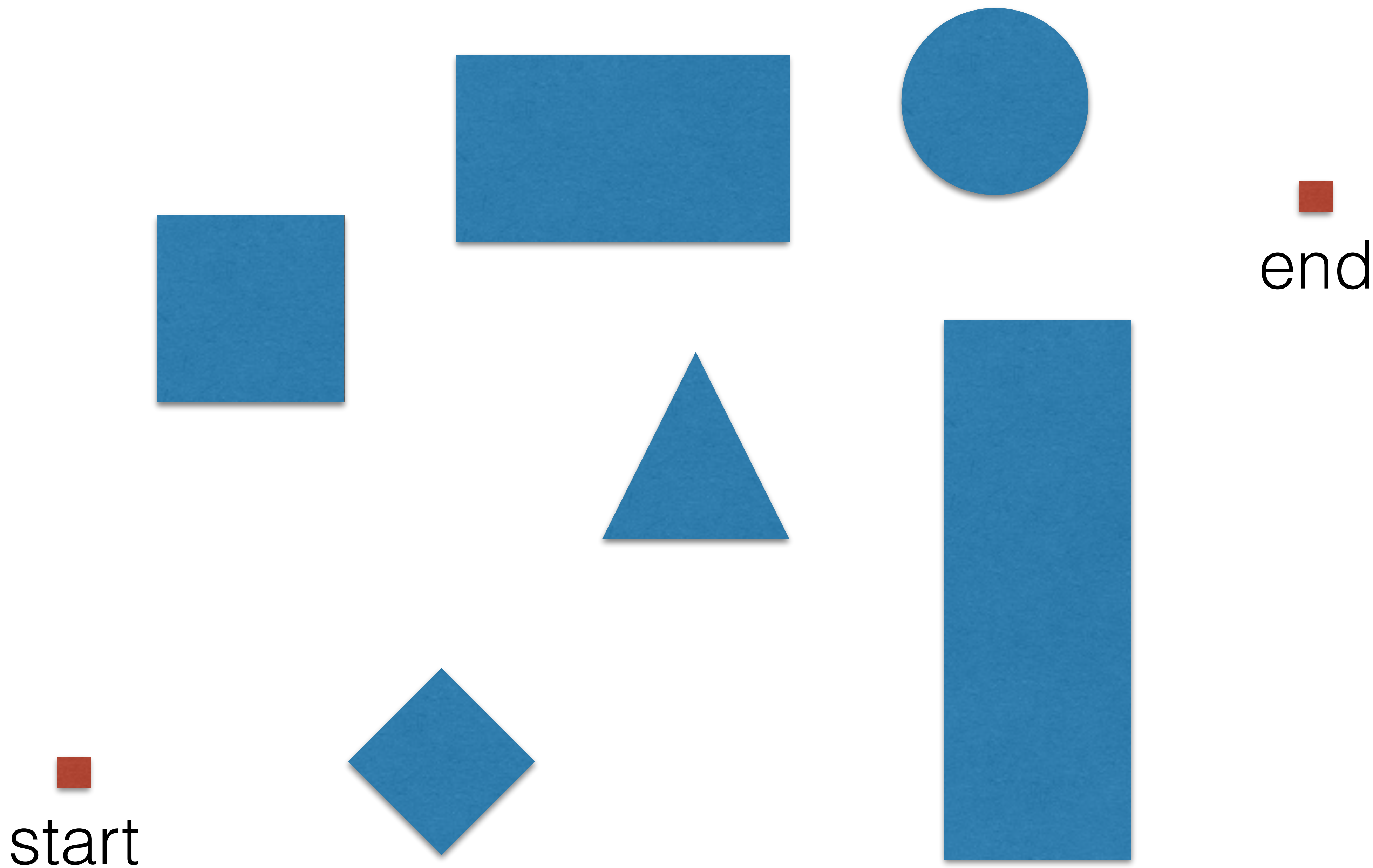
distance from a point to a segment



$$d(x,l) = \min \{d(x,p) \mid p \text{ on } l\}$$

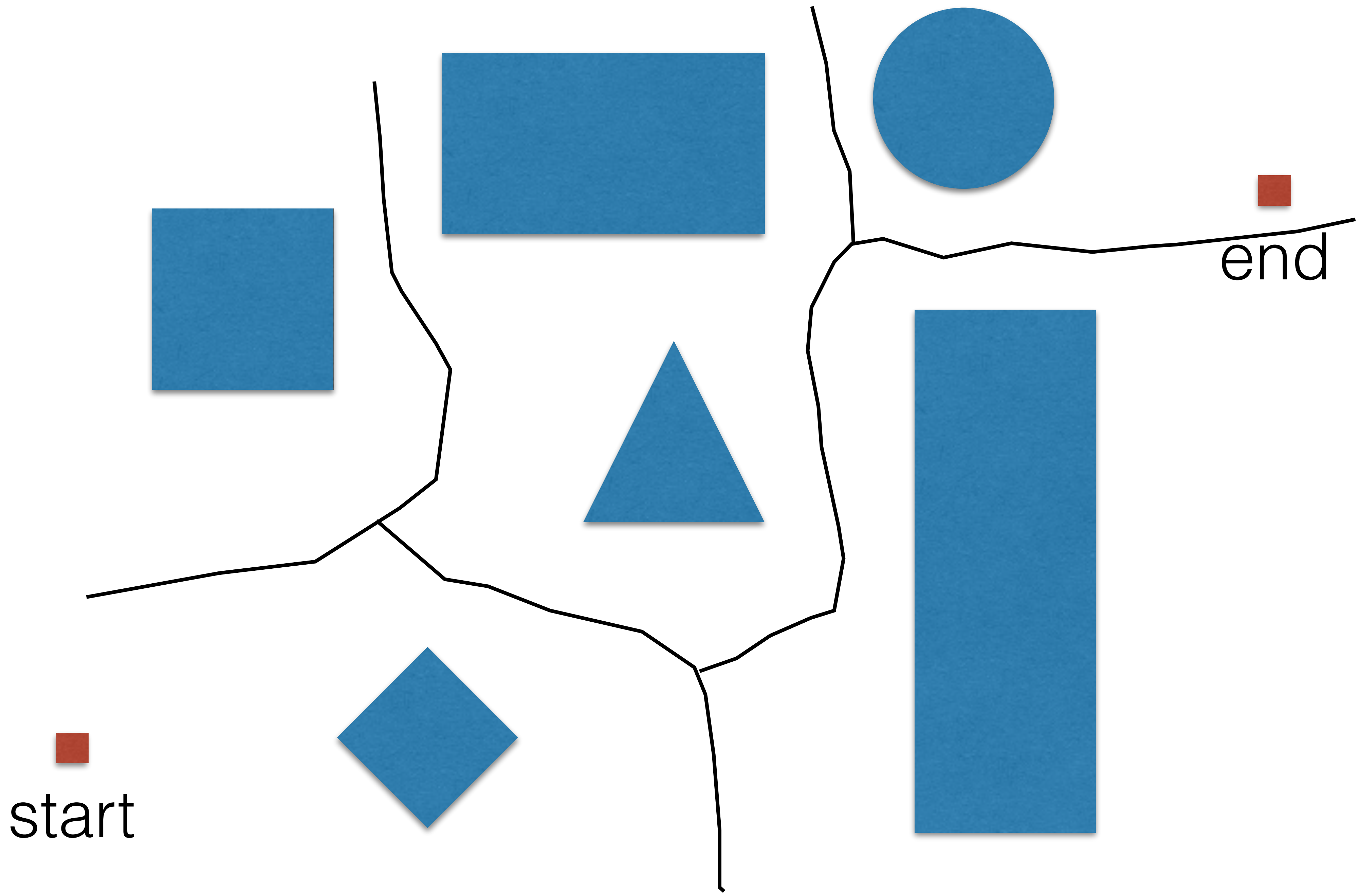
Path planning

To minimize collisions, stay as far away from obstacles as possible.



Path planning

To minimize collisions, stay as far away from obstacles as possible.



Walk on the edges of a Vor(obstacles)

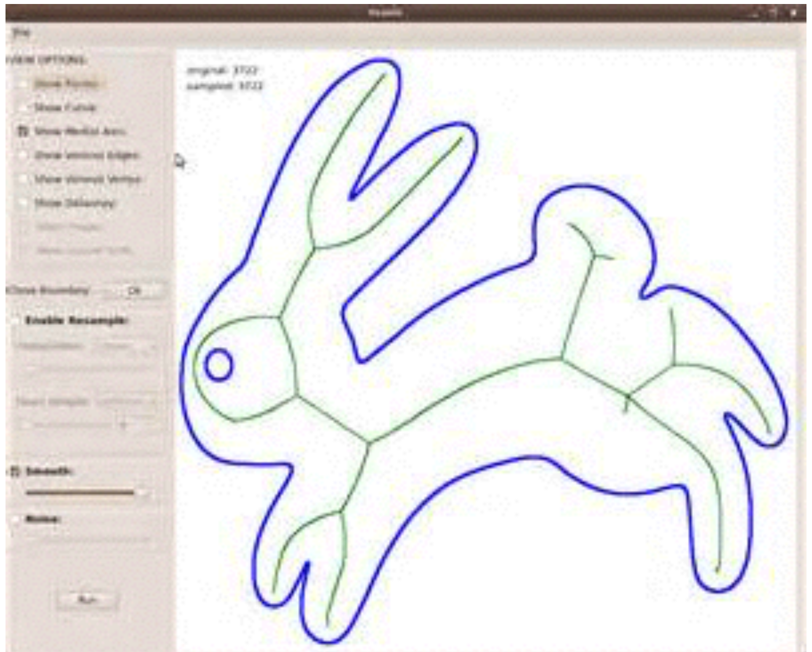
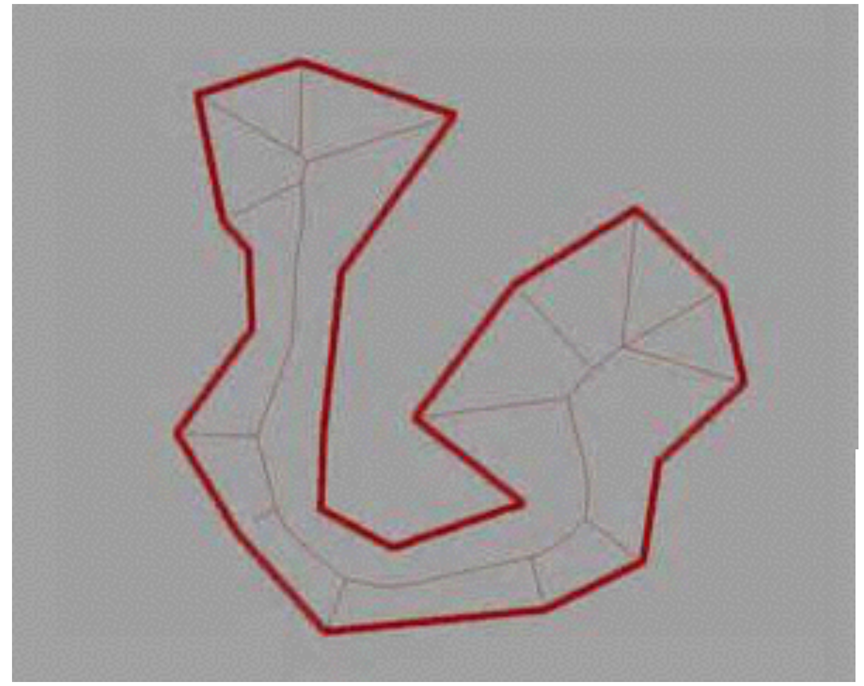
Medial axis

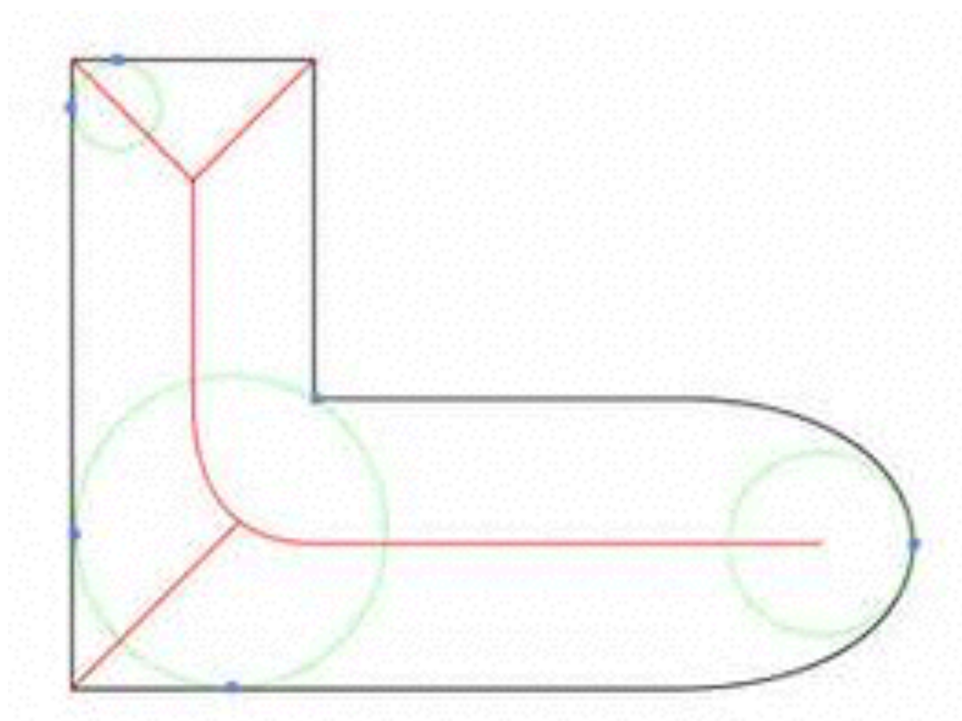
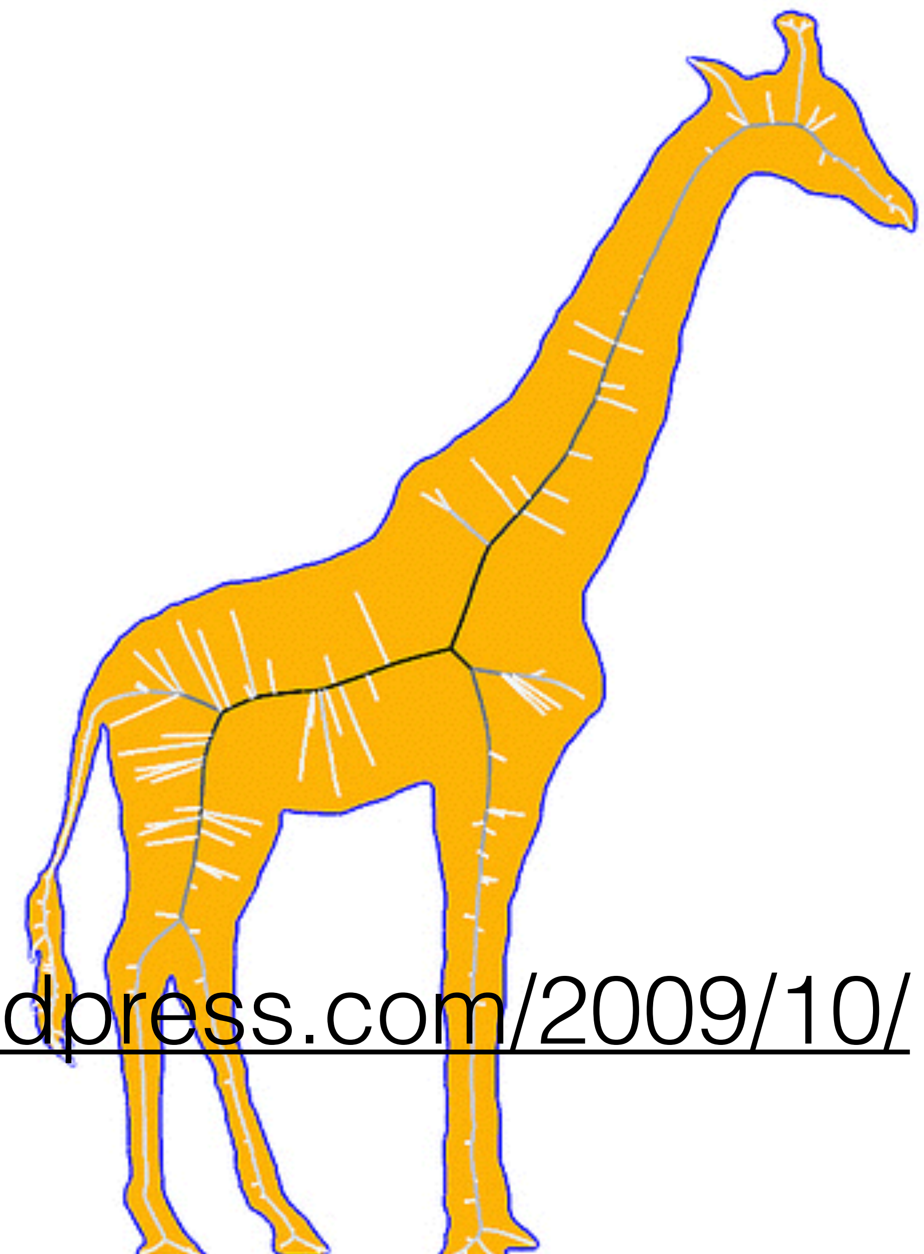
- Compute the Voronoi Diagram of the boundary of a simple (non-intersecting) polygon.
- That is, partition the polygon according to which edge is closest.

Medial axis

- Compute the Voronoi Diagram of the boundary of a simple (non-intersecting) polygon.
- That is, partition the polygon according to which edge is closest.

- Used to study shape
  - vision and image recognition
- Construction
  - medial axis can be constructed in O(n) time for convex polygons
  - In O(n lg n) time for non-convex polygons

https://spacesymmetrystructure.files.wordpress.com/2009/10/medialax.gif

Voronoi diagrams in 3D

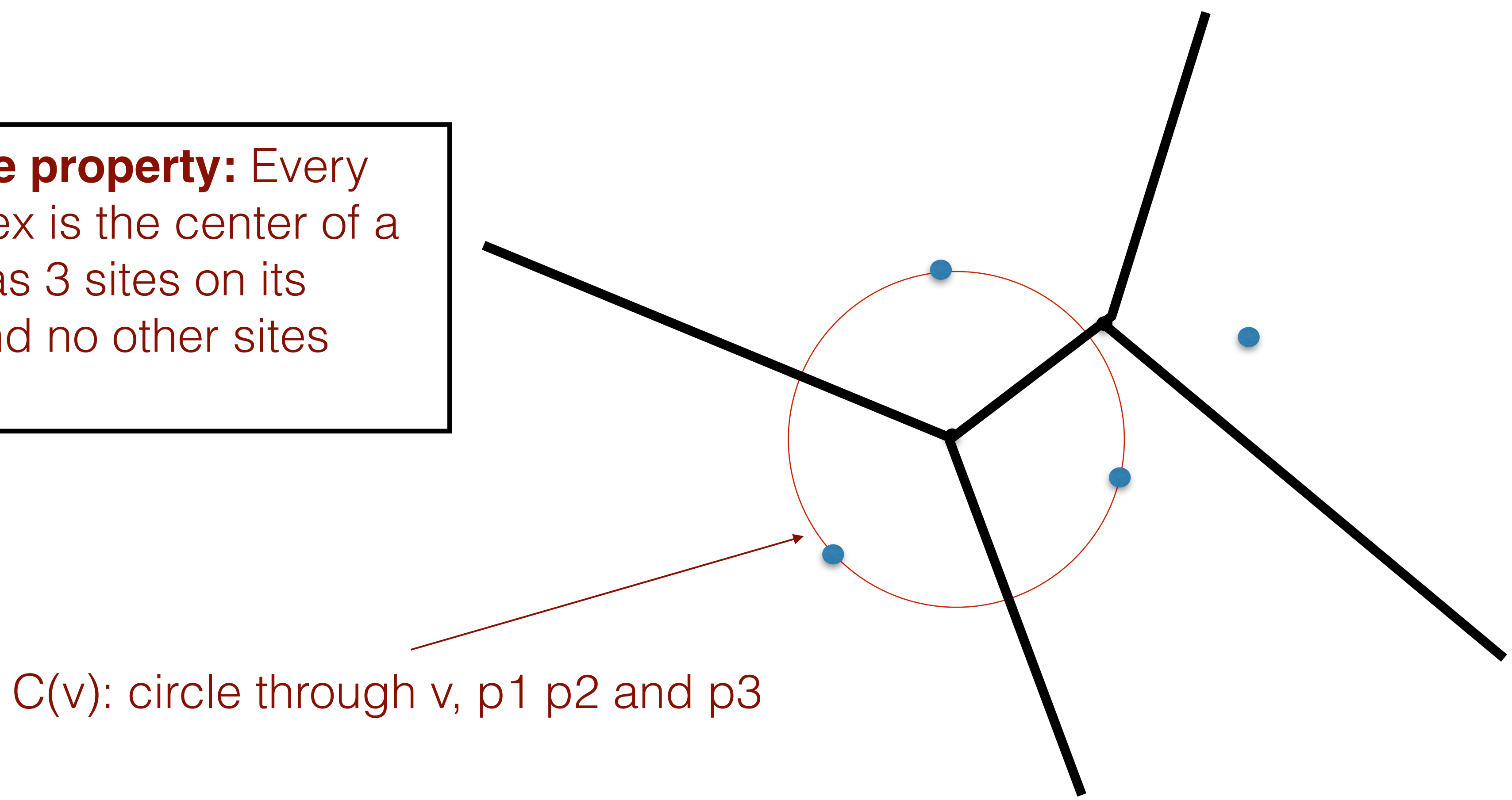
- Partition space according to which site is closest
- Can have $O(n^2)$ size
- There exist algorithm to compute 3D VD in $O(n^2)$ time, which is optimal
- 3D VD are less useful because of their quadratic size

One last property

One last property

Let P = {p_1, p_2,…,p_n} set of points in the plane such that no 4 are co-circular.

Empty circle property: Every Voronoi vertex is the center of a circle that has 3 sites on its boundary and no other sites inside



Theorem: The straight-line dual graph of Vor(P) is a triangulation of P.

The dual of Voronoi is called the Delaunay triangulation.