Voronoi Diagram and Delaunay Triangulation

Chih-Hung Liu

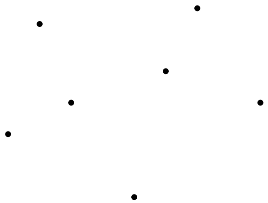
University of Bonn

Outline

1. Voronoi Diagrams and Delaunay Triangulations
   - Properties and Duality
2. 3D geometric transformation
3. Applications

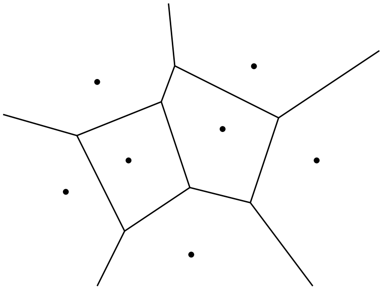
Voronoi Diagram

- Given a set S of n point sites, Voronoi Diagram $V(S)$ is a planar subdivision



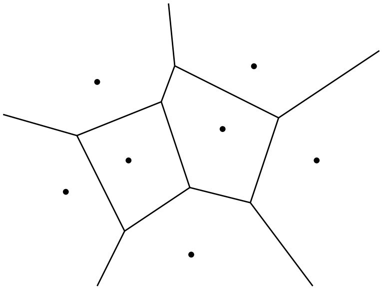
Voronoi Diagram

- Given a set S of n point sites, Voronoi Diagram $V(S)$ is a planar subdivision



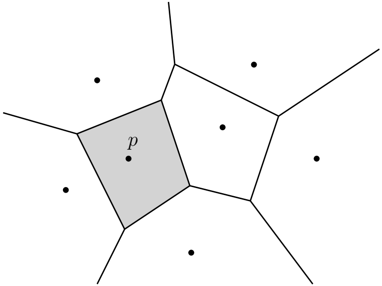
Voronoi Diagram

- Given a set S of n point sites, Voronoi Diagram $V(S)$ is a planar subdivision
  1. Each region contains exactly one site $p \in S$ and is denoted by $\mathrm{VR}(p, S)$.



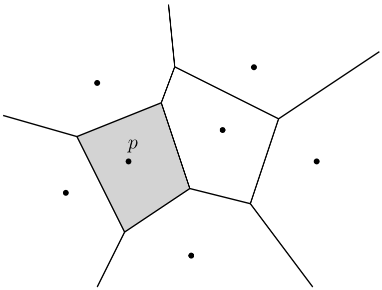
Voronoi Diagram

- Given a set S of n point sites, Voronoi Diagram $V(S)$ is a planar subdivision
  1. Each region contains exactly one site $p \in S$ and is denoted by $\mathrm{VR}(p, S)$.



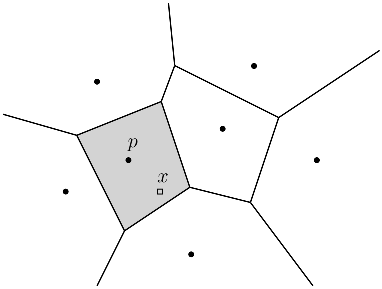
Voronoi Diagram

- Given a set S of n point sites, Voronoi Diagram $V(S)$ is a planar subdivision
  1. Each region contains exactly one site $p \in S$ and is denoted by $\mathrm{VR}(p, S)$.



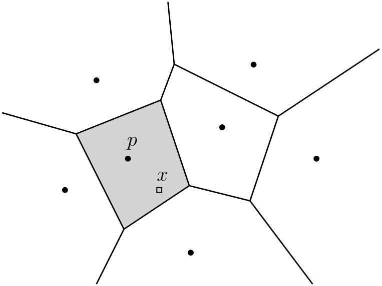
Voronoi Diagram

- Given a set S of n point sites, Voronoi Diagram $V(S)$ is a planar subdivision
  1. Each region contains exactly one site $p \in S$ and is denoted by $\mathrm{VR}(p, S)$.
  2. For each point $x \in \mathrm{VR}(p, S)$, p is its closest site in S.



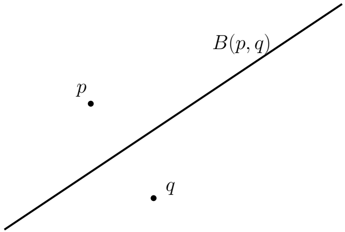
Voronoi Diagram

- Given a set S of n point sites, Voronoi Diagram $V(S)$ is a planar subdivision
  1. Each region contains exactly one site $p \in S$ and is denoted by $\mathrm{VR}(p, S)$.
  2. For each point $x \in \mathrm{VR}(p, S)$, p is its closest site in S.
- $\mathrm{VR}(p, S)$ is the locus of points closer to p than any other site.



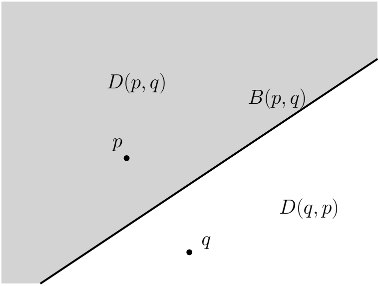
Voronoi Region

- Bisector $\boldsymbol{B(p,q)} = \{x \in R^2 \mid d(x,p) = d(x,q)\}$.



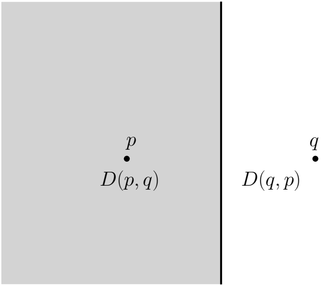
Voronoi Region

- Bisector $\boldsymbol{B(p,q)} = \{x \in R^2 \mid d(x,p) = d(x,q)\}$.
- $\boldsymbol{D(p,q)} = \{x \in R^2 \mid d(x,p) < d(x,q)\}$.
  - Two half-planes $D(p,q)$ and $D(q,p)$ separated by $B(p,q)$.



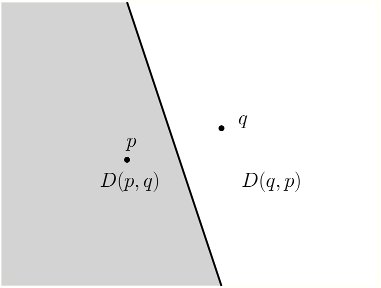
Voronoi Region

- Bisector $\boldsymbol{B(p,q)} = \{x \in R^2 \mid d(x,p) = d(x,q)\}$.
- $\boldsymbol{D(p,q)} = \{x \in R^2 \mid d(x,p) < d(x,q)\}$.
  - Two half-planes $D(p,q)$ and $D(q,p)$ separated by $B(p,q)$.



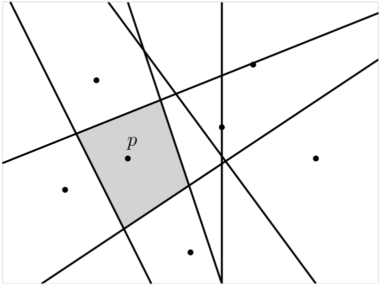
Voronoi Region

- Bisector $\boldsymbol{B(p,q)} = \{x \in R^2 \mid d(x,p) = d(x,q)\}$.
- $\boldsymbol{D(p,q)} = \{x \in R^2 \mid d(x,p) < d(x,q)\}$.
  - Two half-planes $D(p,q)$ and $D(q,p)$ separated by $B(p,q)$.



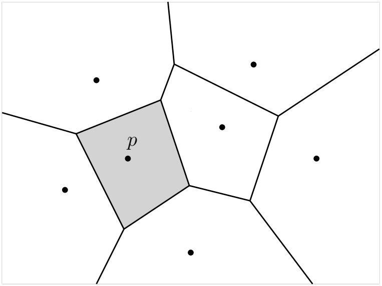
Voronoi Region

- Bisector $\boldsymbol{B(p,q)} = \{x \in R^2 \mid d(x,p) = d(x,q)\}$.
- $\boldsymbol{D(p,q)} = \{x \in R^2 \mid d(x,p) < d(x,q)\}$.
  - Two half-planes $D(p,q)$ and $D(q,p)$ separated by $B(p,q)$.
- $$\mathrm{VR}(p,S) = \bigcap_{q \in S, q \neq p} D(p,q).$$



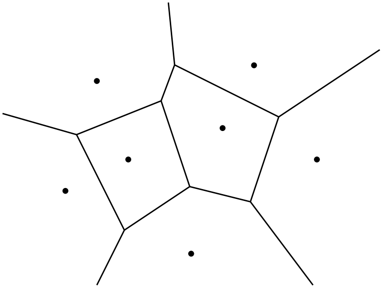
Voronoi Region

- Bisector $\boldsymbol{B(p,q)} = \{x \in R^2 \mid d(x,p) = d(x,q)\}$.
- $\boldsymbol{D(p,q)} = \{x \in R^2 \mid d(x,p) < d(x,q)\}$.
  - Two half-planes $D(p,q)$ and $D(q,p)$ separated by $B(p,q)$.
-
$$\mathrm{VR}(p,S) = \bigcap_{q \in S, q \neq p} D(p,q).$$



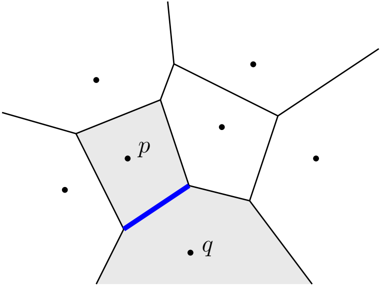
Voronoi Edge and Vertex

- Voronoi Edge
  - Common intersection between two adjacent Voronoi regions $\mathrm{VR}(p, S)$ and $\mathrm{VR}(q, S)$



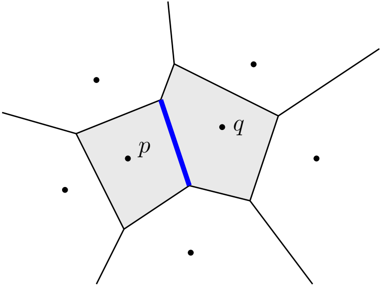
Voronoi Edge and Vertex

- Voronoi Edge
  - Common intersection between two adjacent Voronoi regions VR(p, S) and VR(q, S)
  - A piece of $B(p, q)$



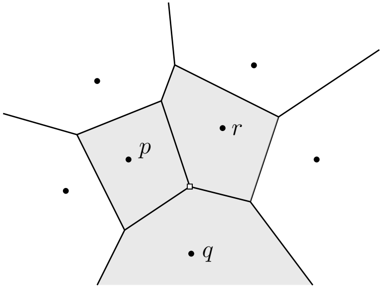
Voronoi Edge and Vertex

- Voronoi Edge
  - Common intersection between two adjacent Voronoi regions $VR(p, S)$ and $VR(q, S)$
  - A piece of $B(p, q)$



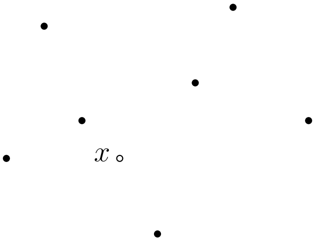
Voronoi Edge and Vertex

- Voronoi Edge
  - Common intersection between two adjacent Voronoi regions $\mathrm{VR}(p, S)$ and $\mathrm{VR}(q, S)$
  - A piece of $B(p, q)$
- Voronoi Vertex
  - Common intersection among more than two Voronoi regions $\mathrm{VR}(p, S)$, $\mathrm{VR}(q, S)$, $\mathrm{VR}(r, S)$, and so on.



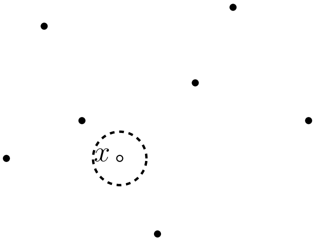
Growing Circle

- Grow a circle from a point *x* on the plane



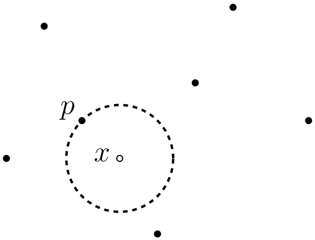
Growing Circle

- Grow a circle from a point *x* on the plane



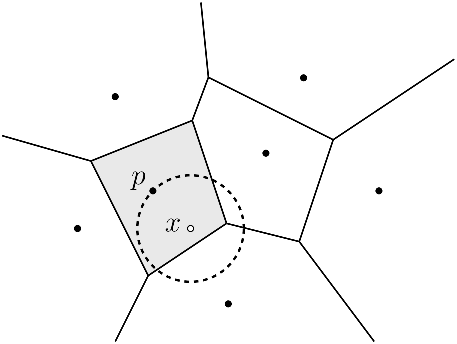
Growing Circle

- Grow a circle from a point x on the plane
  - Hit one site $p \in S \rightarrow x$ belongs to $\mathrm{VR}(p, S)$



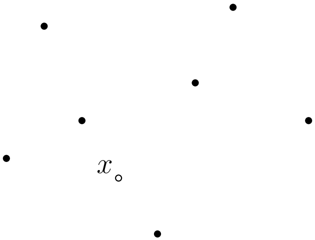
Growing Circle

- Grow a circle from a point x on the plane
  - Hit one site $p \in S \rightarrow x$ belongs to $\mathrm{VR}(p, S)$



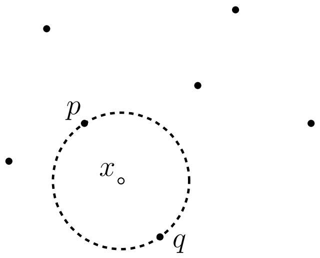
Growing Circle

- Grow a circle from a point x on the plane
  - Hit one site $p \in S \rightarrow x$ belongs to $\mathrm{VR}(p, S)$



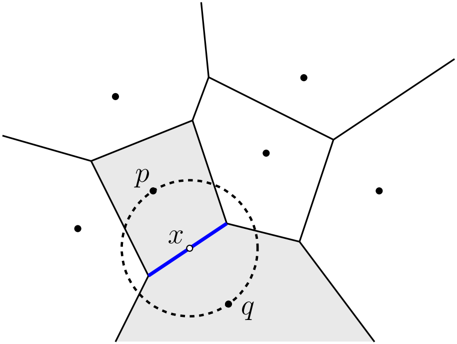
Growing Circle

- Grow a circle from a point x on the plane
  - Hit one site $p \in S \rightarrow x$ belongs to VR(p, S)
  - Hit two sites $p, q \in S \rightarrow x$ belongs to the Voronoi edge between VR(p, S) and VR(q, S)



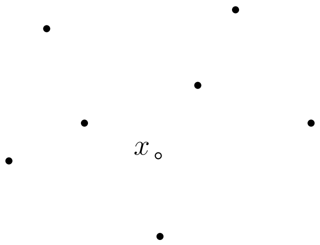
Growing Circle

- Grow a circle from a point x on the plane
  - Hit one site $p \in S \rightarrow x$ belongs to VR(p, S)
  - Hit two sites $p, q \in S \rightarrow x$ belongs to the Voronoi edge between VR(p, S) and VR(q, S)



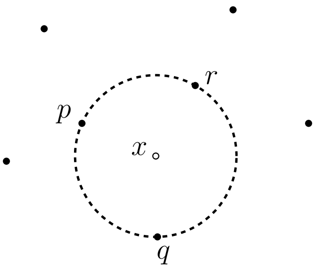
Growing Circle

- Grow a circle from a point x on the plane
  - Hit one site $p \in S \rightarrow x$ belongs to $\mathrm{VR}(p, S)$
  - Hit two sites $p, q \in S \rightarrow x$ belongs to the Voronoi edge between $\mathrm{VR}(p, S)$ and $\mathrm{VR}(q, S)$



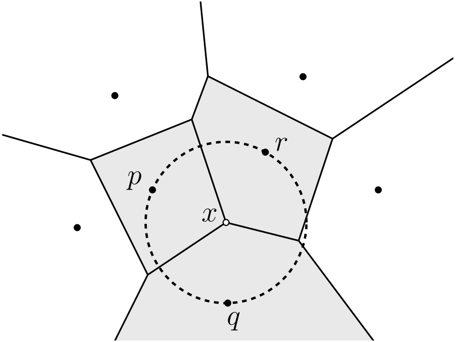
Growing Circle

- Grow a circle from a point x on the plane
  - Hit one site $p \in S \rightarrow x$ belongs to $\mathrm{VR}(p, S)$
  - Hit two sites $p, q \in S \rightarrow x$ belongs to the Voronoi edge between $\mathrm{VR}(p, S)$ and $\mathrm{VR}(q, S)$
  - Hit more than two sites $p, q, r, \ldots \in S \rightarrow x$ is the Voronoi vertex among $\mathrm{VR}(p, S)$, $\mathrm{VR}(q, S)$, $\mathrm{VR}(r, S)$, $\ldots$



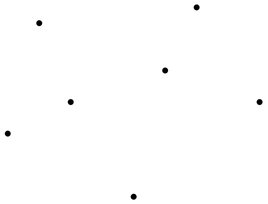
Growing Circle

- Grow a circle from a point x on the plane
  - Hit one site $p \in S \rightarrow x$ belongs to $\mathrm{VR}(p, S)$
  - Hit two sites $p, q \in S \rightarrow x$ belongs to the Voronoi edge between $\mathrm{VR}(p, S)$ and $\mathrm{VR}(q, S)$
  - Hit more than two sites $p, q, r, \ldots \in S \rightarrow x$ is the Voronoi vertex among $\mathrm{VR}(p, S)$, $\mathrm{VR}(q, S)$, $\mathrm{VR}(r, S)$, ...



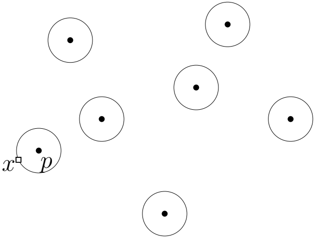
Wavefront Model (Growth Model)

- Grow circles from $\forall p \in S$ at unit speed



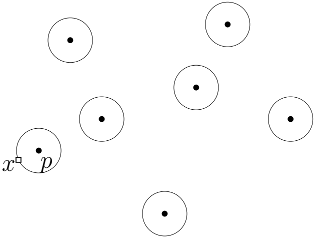
Wavefront Model (Growth Model)

- Grow circles from $\forall p \in S$ at unit speed



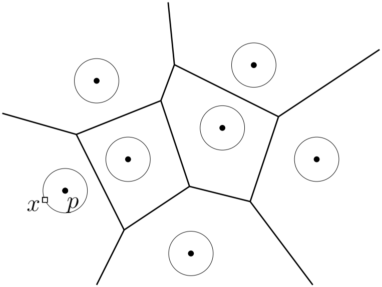
Wavefront Model (Growth Model)

- Grow circles from $\forall p \in S$ at unit speed
  - $x \in R^2$ is first hit by a circle from $p \to x$ belongs to $\mathrm{VR}(p, S)$



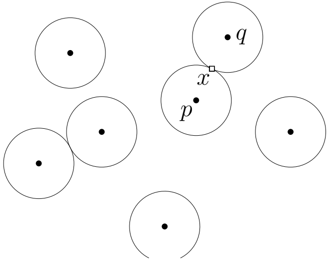
Wavefront Model (Growth Model)

- Grow circles from $\forall p \in S$ at unit speed
  - $x \in R^2$ is first hit by a circle from $p \to x$ belongs to $\mathrm{VR}(p, S)$



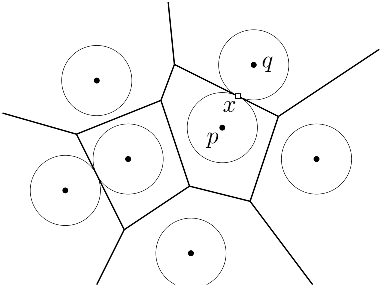
Wavefront Model (Growth Model)

- Grow circles from $\forall p \in S$ at unit speed
  - $x \in R^2$ is first hit by a circle from $p \rightarrow x$ belongs to $\mathrm{VR}(p, S)$
  - $x \in R^2$ is first hit by two circles from p and $q \rightarrow x$ belongs to a Voronoi edge between $\mathrm{VR}(p, S)$ and $\mathrm{VR}(q, S)$



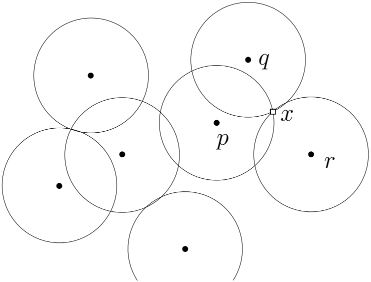
Wavefront Model (Growth Model)

- Grow circles from $\forall p \in S$ at unit speed
  - $x \in R^2$ is first hit by a circle from $p \rightarrow x$ belongs to VR(p, S)
  - $x \in R^2$ is first hit by two circles from p and $q \rightarrow x$ belongs to a Voronoi edge between VR(p, S) and VR(q, S)



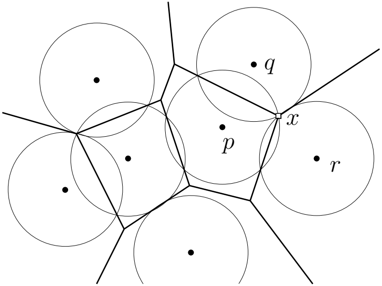
Wavefront Model (Growth Model)

- Grow circles from $\forall p \in S$ at unit speed
  - $x \in R^2$ is first hit by a circle from $p \rightarrow x$ belongs to $\mathrm{VR}(p, S)$
  - $x \in R^2$ is first hit by two circles from p and $q \rightarrow x$ belongs to a Voronoi edge between $\mathrm{VR}(p, S)$ and $\mathrm{VR}(q, S)$
  - $x \in R^2$ is first hit by three circles from p, q, and $r \rightarrow x$ is a Voronoi vertex among $\mathrm{VR}(p, S)$, $\mathrm{VR}(q, S)$ and $\mathrm{VR}(r, S)$



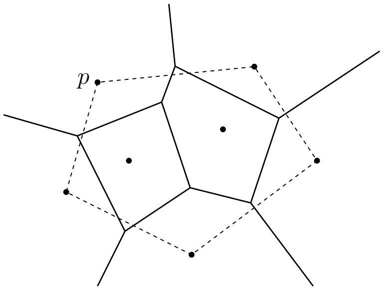
Wavefront Model (Growth Model)

- Grow circles from $\forall p \in S$ at unit speed
  - $x \in R^2$ is first hit by a circle from $p \rightarrow x$ belongs to $\mathrm{VR}(p, S)$
  - $x \in R^2$ is first hit by two circles from p and $q \rightarrow x$ belongs to a Voronoi edge between $\mathrm{VR}(p, S)$ and $\mathrm{VR}(q, S)$
  - $x \in R^2$ is first hit by three circles from p, q, and $r \rightarrow x$ is a Voronoi vertex among $\mathrm{VR}(p, S)$, $\mathrm{VR}(q, S)$ and $\mathrm{VR}(r, S)$



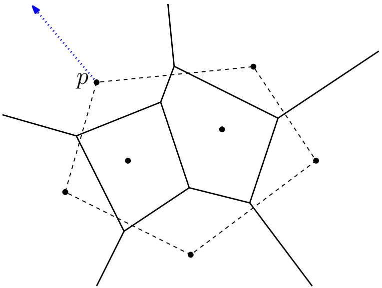
Unbounded Region

- VR(p, S) is unbounded if and only if p is a vertex of the convex hull of S.



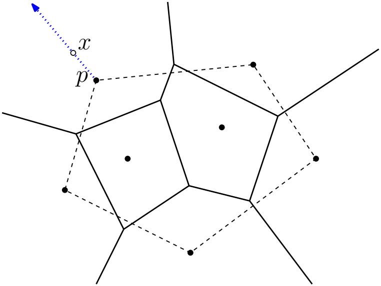
Unbounded Region

- VR(p, S) is unbounded if and only if p is a vertex of the convex hull of S.
  - Consider the exterior angle bisector of p



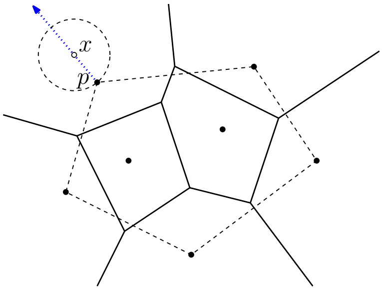
Unbounded Region

- VR(p, S) is unbounded if and only if p is a vertex of the convex hull of S.
  - Consider the exterior angle bisector of p
  - For any point x on the bisector, x belongs to VR(p, S)



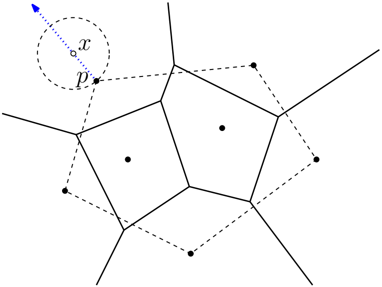
Unbounded Region

- $\mathrm{VR}(p, S)$ is unbounded if and only if p is a vertex of the convex hull of S.
  - Consider the exterior angle bisector of p
  - For any point x on the bisector, x belongs to $\mathrm{VR}(p, S)$



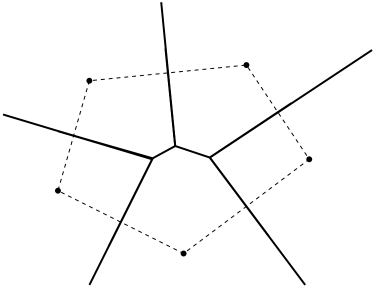
Unbounded Region

- VR(p, S) is unbounded if and only if p is a vertex of the convex hull of S.
  - Consider the exterior angle bisector of p
  - For any point x on the bisector, x belongs to VR(p, S)
  - The bisector extends to the infinity.



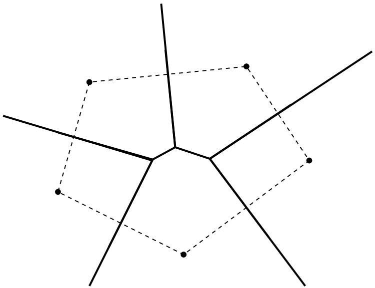
Unbounded Region

- VR(p, S) is unbounded if and only if p is a vertex of the convex hull of S.
  - Consider the exterior angle bisector of p
  - For any point x on the bisector, x belongs to VR(p, S)
  - The bisector extends to the infinity.
- If S is in convex position, $V(S)$ is a tree.



Unbounded Region

- VR(p, S) is unbounded if and only if p is a vertex of the convex hull of S.
  - Consider the exterior angle bisector of p
  - For any point x on the bisector, x belongs to VR(p, S)
  - The bisector extends to the infinity.
- If S is in convex position, $V(S)$ is a tree.
- An unbounded Voronoi edge corresponds to a hull edge.



Voronoi Diagram (Mathematic Definition)

- Voronoi Diagram $V(S)$

$$V(S) = R^2 \setminus (\bigcup_{p \in S} \mathrm{VR}(p, S)) = \bigcup_{p \in S} \partial \mathrm{VR}(p, S)$$

  - $\partial \mathrm{VR}(p, S)$ is the boundary of $\mathrm{VR}(p, S)$
    - $\partial \mathrm{VR}(p, S) \not\subset \mathrm{VR}(p, S)$
  - $V(S)$ is the union of all the Voronoi edges
- Voronoi Edge e between $\mathrm{VR}(p, S)$ and $\mathrm{VR}(q, S)$

$$e = \partial \mathrm{VR}(p, S) \cap \partial \mathrm{VR}(q, S)$$

- Voronoi Vertex v among $\mathrm{VR}(p, S)$, $\mathrm{VR}(q, S)$, and $\mathrm{VR}(r, S)$

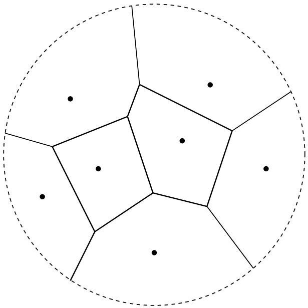
$$v = \partial \mathrm{VR}(p, S) \cap \partial \mathrm{VR}(q, S) \cap \partial \mathrm{VR}(r, S)$$

Complexity of $V(S)$

Theorem

$V(S)$ has $O(n)$ edges and vertices. The average number of edges of a Voronoi region is less than 6.

- Add a large curve Γ
  - Γ only passes through unbounded edges of $V(S)$
  - Cut unbounded pieces outside Γ
  - One additional face and several edges and vertices.



Complexity of $V(S)$

Theorem

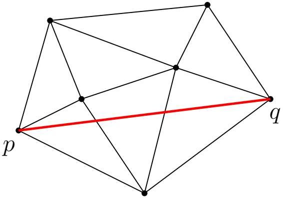
$V(S)$ has $O(n)$ edges and vertices. The average number of edges of a Voronoi region is less than 6.

- Euler's Polyhedron Formula: $v - e + f = 1 + c$
  - v: # of vertices, e: # of edges, f: # of faces, and c: # number of connected components.
- An edge has two endpoints, and a vertex is incident to at least three edges.
  - $3v \leq 2e \rightarrow v \leq 2e/3$
- $f = n + 1$ and $c = 1$
  - $v = 1 + c + e - f = e + 1 - n \leq 2e/3 \rightarrow e \leq 3n - 3$
  - $e = v + f - 1 - c = v + n - 1 \geq 3v/2 \rightarrow v \leq 2n - 2$
- Average number of edges of a region $\leq (6n - 6)/n < 6$

Triangulation

Definition

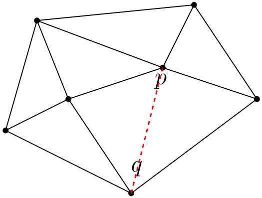
Given a set S of points on the plane, a triangulation is maximal collection of non-crossing line segments among S.



Crossing ($\overline{pq}$)

Definition

Given a set S of points on the plane, a triangulation is maximal collection of non-crossing line segments among S.

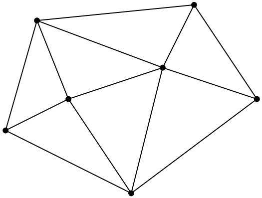


Not Maximal ($\overline{pq}$ is allowable)

Triangulation

Definition

Given a set S of points on the plane, a triangulation is maximal collection of non-crossing line segments among S.

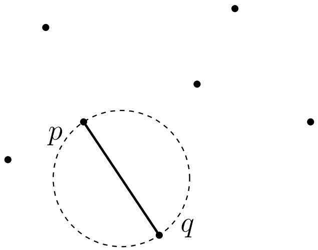


Triangulation

Delaunay Edge

Definition

An edge $\overline{pq}$ is called **Delaunay** if there exists a circle passing through p and q and containing no other point in its interior.

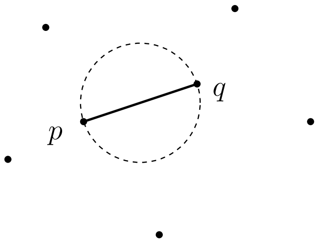


$\overline{pq}$ is **Delaunay**

Delaunay Edge

Definition

An edge $\overline{pq}$ is called **Delaunay** if there exists a circle passing through p and q and containing no other point in its interior.

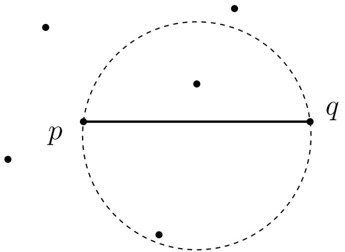


$\overline{pq}$ is **Delaunay**

Delaunay Edge

Definition

An edge $\overline{pq}$ is called **Delaunay** if there exists a circle passing through p and q and containing no other point in its interior.

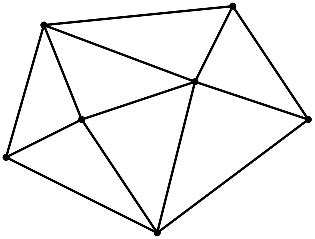


$\overline{pq}$ is **NOT** Delaunay

Delaunay Triangulation

Definition

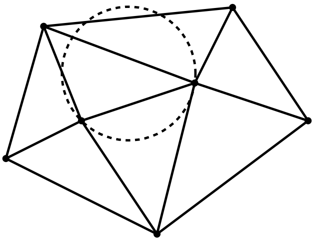
A **Delaunay Triangulation** is a triangulation whose edges are all Delaunay.



Delaunay Triangulation

Definition

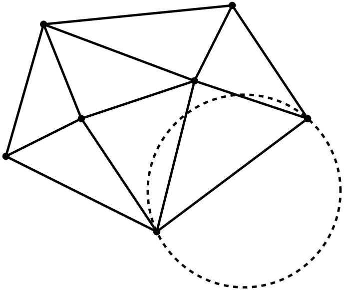
A **Delaunay Triangulation** is a triangulation whose edges are all Delaunay.



Delaunay Triangulation

Definition

A **Delaunay Triangulation** is a triangulation whose edges are all Delaunay.

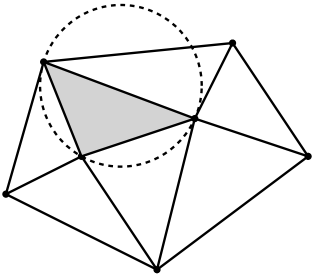


Delaunay Triangulation

Definition

A **Delaunay Triangulation** is a triangulation whose edges are all Delaunay.

- For each face, there exists a circle passing all its vertices and containing no other point.



General Position Assumption

1. No more than two point sites are colinear

General Position Assumption

1. No more than two point sites are colinear
   - $V(S)$ is **connected**

General Position Assumption

1. No more than two point sites are colinear
   - $V(S)$ is **connected**
2. No more than three point sites are cocircular
   (At most three points are on the same circle)

General Position Assumption

1. No more than two point sites are colinear
   - $V(S)$ is **connected**
2. No more than three point sites are cocircular
   (At most three points are on the same circle)
   - degree of each Voronoi vertex is exactly **3**.

General Position Assumption

1. No more than two point sites are colinear
   - $V(S)$ is **connected**
2. No more than three point sites are cocircular
   (At most three points are on the same circle)
   - degree of each Voronoi vertex is exactly **3**.
   - Each face of the Delaunay triangulation is a **triangle**.

General Position Assumption

1. No more than two point sites are colinear
   - $V(S)$ is **connected**
2. No more than three point sites are cocircular
   (At most three points are on the same circle)
   - degree of each Voronoi vertex is exactly **3**.
   - Each face of the Delaunay triangulation is a **triangle**.

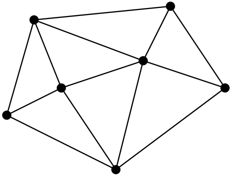
- There is a unique Delaunay triangulation.

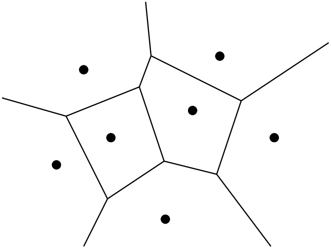
Duality

Theorem

Under the general position assumption, the Delaunay triangulation is a dual graph of the Voronoi diagram.

- A site $p \leftrightarrow$ a Voronoi region $VR(p, S)$



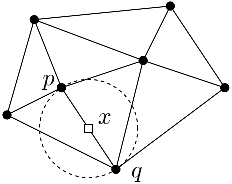


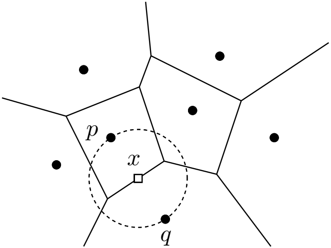
Duality

Theorem

Under the general position assumption, the Delaunay triangulation is a dual graph of the Voronoi diagram.

- A site $p \leftrightarrow$ a Voronoi region $\mathrm{VR}(p, S)$
- A Delaunay edge $\overline{pq} \leftrightarrow$ a Voronoi edge between $\mathrm{VR}(p, S)$ and $\mathrm{VR}(q, S)$



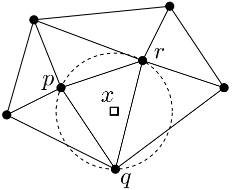


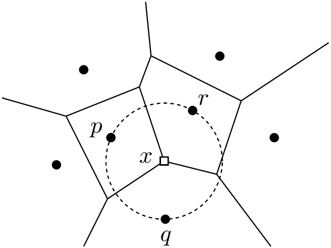
Duality

Theorem

Under the general position assumption, the Delaunay triangulation is a dual graph of the Voronoi diagram.

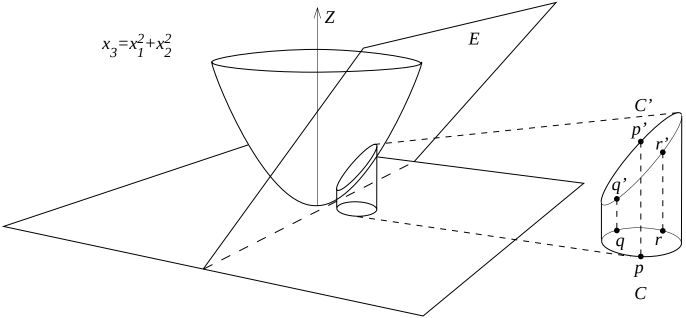
- A site p $\leftrightarrow$ a Voronoi region $VR(p, S)$
- A Delaunay edge $\overline{pq}$ $\leftrightarrow$ a Voronoi edge between $VR(p, S)$ and $VR(q, S)$
- A Delaunay triangle Δpqr $\leftrightarrow$ a Voronoi vertex among $VR(p, S)$, $VR(q, S)$ and $VR(r, S)$





Geometric Transformation from 2D to 3D

- A paraboloid $P = \{(x_1, x_2, x_3) \mid x_1^2 + x_2^2 = x_3\}$ in 3D
- For a point $x = (x_1, x_2)$ in 2D, $x' = (x_1, x_2, x_1^2 + x_2^2)$ is its lifted image in 3D
  - $x' \leftarrow$ vertical projection from x to P
- For a set A of points in 2D, its lifted image $A' = \{x' = (x_1, x_2, x_1^2 + x_2^2) \mid x = (x_1, x_2) \in A\}$



Circle in 2D $\leftrightarrow$ Planar Curve in P

Lemma

Let C be a circle in the plane. Then C' is a planar curve on the paraboloid P

- C is given by $r^2 = (x_1 - c_1)^2 + (x_2 - c_2)^2$
  - $r^2 = x_1^2 + x_2^2 - 2x_1c_1 - 2x_2c_2 + c_1^2 + c_2^2$
- C' satisfies $x_1^2 + x_2^2 = x_3$
- Substituting $x_1^2 + x_2^2$ by x_3, we obtain a plane E

$$x_3 - 2x_1c_1 - 2x_2c_2 + c_1^2 + c_2^2 - r^2 = 0$$

- $C' = P \cap E$
- Intersection between E and P is a planar curve

Lower Convex Hull

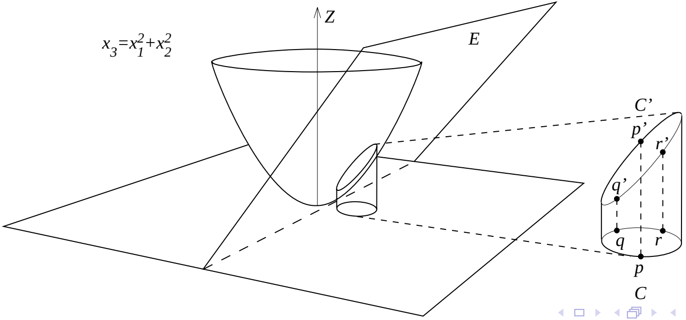
- S' on $P \rightarrow S'$ in convex position
- Each point of S' is a vertex of $conv(S')$
- Lower convex hull $lconv(S')$ of S' is the part of $conv(S')$ visible from $x_3 = -\infty$

Duality between $\mathrm{DT}(S)$ and $lconv(S')$ (1)

Theorem

The Delaunay triangulation $\mathrm{DT}(S)$ equals to the vertical projection onto the x_1x_2-plane of the lower convex hull of S'

- $p, q, r \in S$. C: circumcircle of p, q, r
- C' lies on a plane E defined by p', q', r'
- a point x inside $C \leftrightarrow$ lifted image x' below E



Duality between $\mathrm{DT}(S)$ and $lconv(S')$ (2)

Theorem

The Delaunay triangulation $\mathrm{DT}(S)$ equals to the vertical projection onto the x_1x_2-plane of the lower convex hull of S'

- p, q, r defines a triangle of $\mathrm{DT}(S)$
  $\leftrightarrow$ no point of S in C defined by p, q, r
  $\leftrightarrow$ no point of S' below E defined by p', q', r'
  $\leftrightarrow$ p', q', r' defines a facet of $lconv(S')$
- Computing a convex hull in 3D takes $O(n \log n)$ time
  - $V(S)$ in $O(n \log n)$ time

Another Viewpoint of paraboloid

- For each $s = (s_1, s_2) \in S$, a paraboloid

$$P_s = \{(x_1, x_2, x_3) \mid x_3 = (x_1 - s_1)^2 + (x_2 - s_2)^2\}$$

  - For each $x = (\sigma_1, \sigma_2)$ in $x_1 x_2$ plane, vertical distance from x to P_s is $d(x, s)^2$
- Opaque and of pairwise different colors
- Looking from $x_3 = -\infty$ upward $\rightarrow V(S)$
- Vertical from x upward first hits $P_s \rightarrow x \in \mathrm{VR}(p, S)$
- $P_s \cap P_t \rightarrow B(s, t)$
- Lower envelope of $\bigcup_{s \in S} P_s \rightarrow V(S)$

Wavefront model revisited

- $P_s = \{(x_1, x_2, x_3) | x_3 = f((x_1 - s_1)^2 + (x_2 - s_2)^2)\}$
  - f is a strictly increasing function
  - Lower envelope of $\bigcup_{s \in S} P_s \to V(S)$

Wavefront model revisited

- $P_s = \{(x_1, x_2, x_3) | x_3 = f((x_1 - s_1)^2 + (x_2 - s_2)^2)\}$
  - f is a strictly increasing function
  - Lower envelope of $\bigcup_{s \in S} P_s \rightarrow V(S)$
- $f(x) = \sqrt{x} = \sqrt{(x_1 - s_1)^2 + (x_2 - s_2)^2}$

Wavefront model revisited

- $P_s = \{(x_1, x_2, x_3) | x_3 = f((x_1 - s_1)^2 + (x_2 - s_2)^2)\}$
  - f is a strictly increasing function
  - Lower envelope of $\bigcup_{s \in S} P_s \to V(S)$
- $f(x) = \sqrt{x} = \sqrt{(x_1 - s_1)^2 + (x_2 - s_2)^2}$
  - Cones of slope $45°$ with apices at sites $s \in S$

Wavefront model revisited

- $P_s = \{(x_1, x_2, x_3) | x_3 = f((x_1 - s_1)^2 + (x_2 - s_2)^2)\}$
  - f is a strictly increasing function
  - Lower envelope of $\bigcup_{s \in S} P_s \to V(S)$
- $f(x) = \sqrt{x} = \sqrt{(x_1 - s_1)^2 + (x_2 - s_2)^2}$
  - Cones of slope $45°$ with apices at sites $s \in S$
- Expanding circles C_s from sites $s \in S$ at equal unit speed
  - time t = radius r
  - $r^2 = (x_1 - s_1)^2 + (x_2 - s_2)^2$
  - x_3 = $\sqrt{(x_1 - s_1)^2 + (x_2 - s_2)^2}$ = radius = time

Wavefront model revisited

- $P_s = \{(x_1, x_2, x_3) | x_3 = f((x_1 - s_1)^2 + (x_2 - s_2)^2)\}$
  - f is a strictly increasing function
  - Lower envelope of $\bigcup_{s \in S} P_s \rightarrow V(S)$
- $f(x) = \sqrt{x} = \sqrt{(x_1 - s_1)^2 + (x_2 - s_2)^2}$
  - Cones of slope 45° with apices at sites $s \in S$
- Expanding circles C_s from sites $s \in S$ at equal unit speed
  - time t = radius r
  - $r^2 = (x_1 - s_1)^2 + (x_2 - s_2)^2$
  - $x_3 = \sqrt{(x_1 - s_1)^2 + (x_2 - s_2)^2}$ = radius = time
- x first hit by $C_s \leftrightarrow$ upward vertical projection from x first hit P_s

Post Office Problem

Post Office Problem

Given a query point $x = (x_1, x_2)$, answer the closest post office p among S

- $p \in S$ is the closest post office to x iff x belong to $\mathrm{VR}(p, S)$
  1. Compute $V(S)$ in $O(n \log n)$ time
  2. Construct a point location data structure for $V(S)$ in $O(n \log n)$ time
- Answer each query in $O(\log n)$ time after $O(n \log n)$-time preprocessing
  - Locating x in $\mathrm{VR}(p, S)$ takes $O(\log n)$ time

Nearest Neighbors

Nearest Neighbors

Given a set S of points, for each $s \in S$, compute its nearest neighbor among $S \setminus \{s\}$

Lemma

t is the nearest neighbor of s only if (s, t) is an edge of $\mathrm{DT}(S)$

- A circle growing from s will first hit t
  $\rightarrow$ There is a circle touching s and t but empty of $S \setminus \{s, t\}$

Theorem

The nearest neighbors can be computed in $O(n \log n)$ time

- $\mathrm{DT}(S)$ can be computed in $O(n \log n)$ time
- Since $|\mathrm{DT}(S)|$ is $O(n)$, computing the nearest neighbor from $\mathrm{DT}(S)$ takes $O(n)$ time

Largest Empty Circle

Largest Empty Circle

Given a set S of points inside a conex polygon A, find the largest circle C whose center is located in A and which enclose no point of S.

Lemma

C must touch two or three points of S

- touch no site: expand it until it touches one sites p
- touches one site p: expand it until it touches p and q
- touches p and q: expand it along $B(p, q)$ until
  1. the center of C hit the boundary of A
  2. C hit the third site r

Theorem

The center of C is either a Voronoi vertex of $V(S)$ or the intersection between a Voronoi edge and the boundary of A

Minimum Spanning Tree

Given a set S of points, compute a tree $\mathrm{MST}(S)$ connecting all sites of S with the minimum total length

Minimum Spanning Tree

Given a set S of points, compute a tree $\mathrm{MST}(S)$ connecting all sites of S with the minimum total length

Cut Property

If S is partitioned into A and B, the shortest edge (a, b) satisfying $a \in A$ and $b \in B$ is an edge of $\mathrm{MST}(S)$

Minimum Spanning Tree

Minimum Spanning Tree

Given a set S of points, compute a tree $\mathrm{MST}(S)$ connecting all sites of S with the minimum total length

Cut Property

If S is partitioned into A and B, the shortest edge (a, b) satisfying $a \in A$ and $b \in B$ is an edge of $\mathrm{MST}(S)$

Theorem

An edge of $\mathrm{MST}(S)$ is an edge of $\mathrm{DT}(S)$

- Each shortest edge (a, b), $a \in A$ and $b \in B$, is an edge of $\mathrm{DT}(S)$.
- The circle whose diameter $\overline{ab}$ enclose no other sites
  - If it contains a site $a' \in A$, $d(a', b) < d(a, b)$

Thank You!!

Divide and Conquer

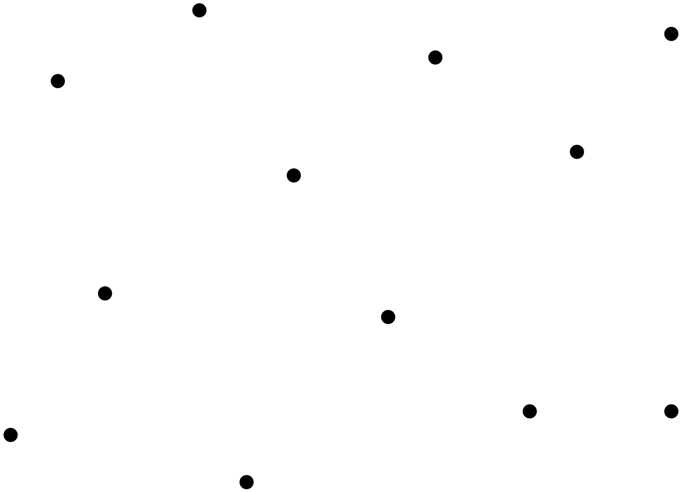
- Basic Steps
  1. Divide an instance into c equal-size sub-instances
     - If the instance is extremely small, solve it directly instead
  2. Recursively compute the sub-solution for each sub-instance
     - Until a sub-instance can be solved in $O(1)$ time
  3. Merge all the c sub-solutions into one solution
- If both Divide and Merge take linear time,

$$T(n) = cT(n/c) + O(n) => O(n \log n)$$

  - The i^{th} level has c^i parts and each part has n/c^i elements
  - A level takes $c^i \times O(n/c^i) = O(n)$ time
  - $O(\log n)$ levels

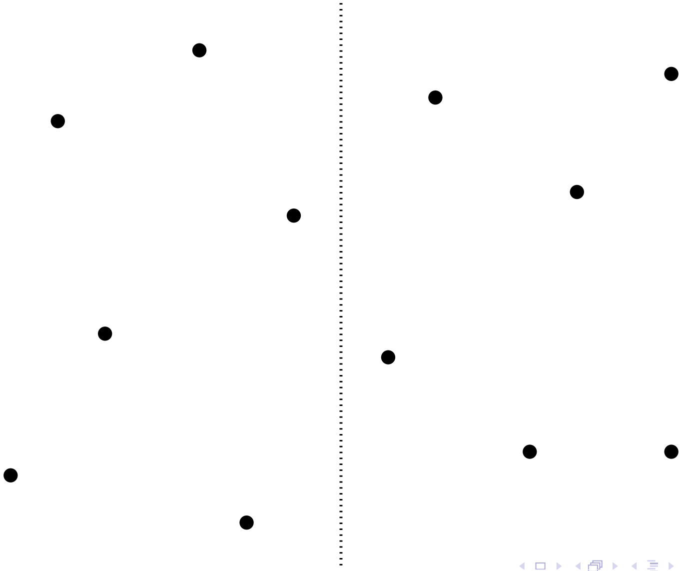
Example: Convex Hull

- $n = 12$ and $c = 2$



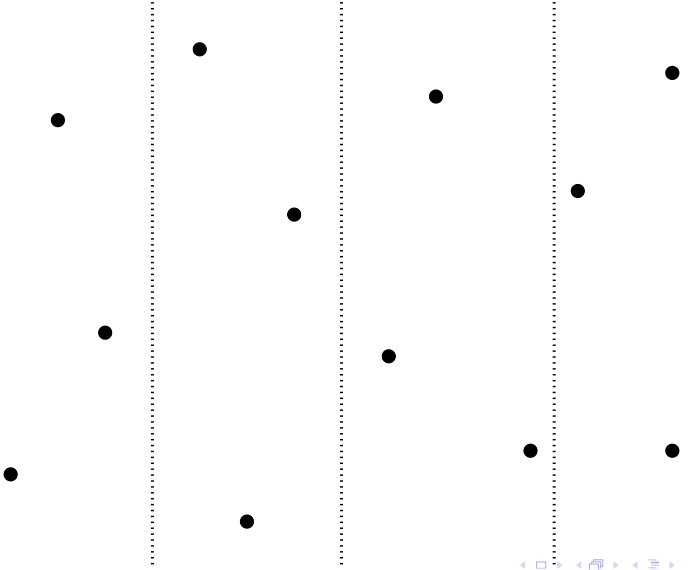
Example: Convex Hull

- $n = 12$ and $c = 2$



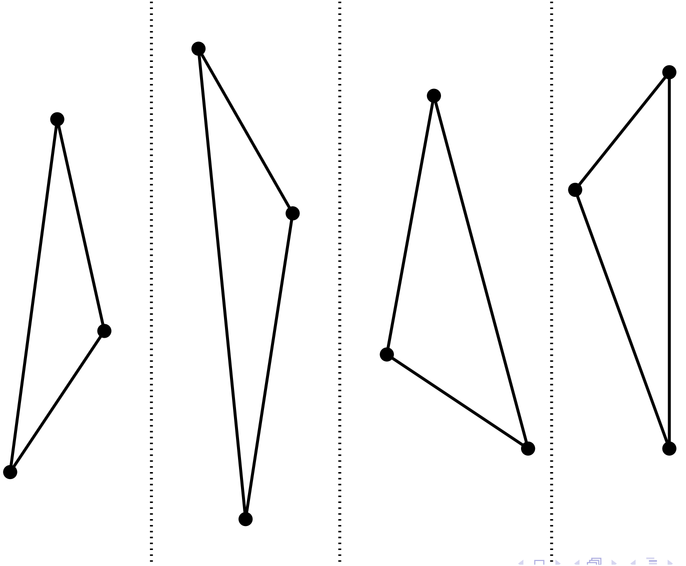
Example: Convex Hull

- $n = 12$ and $c = 2$



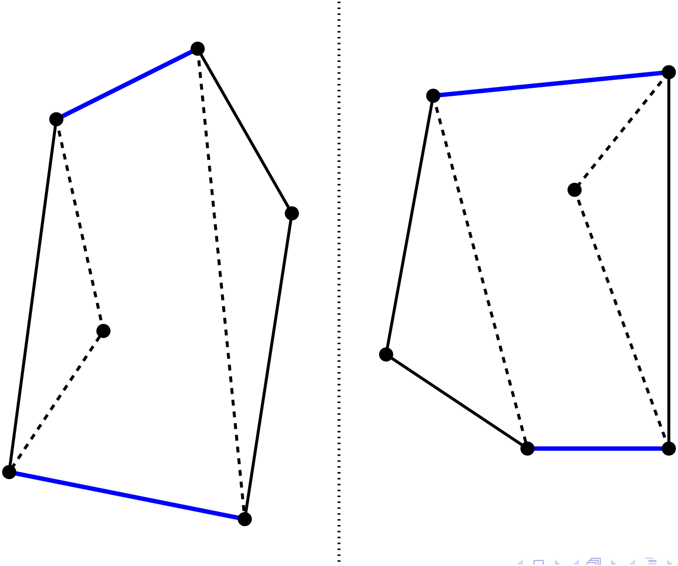
Example: Convex Hull

- $n = 12$ and $c = 2$



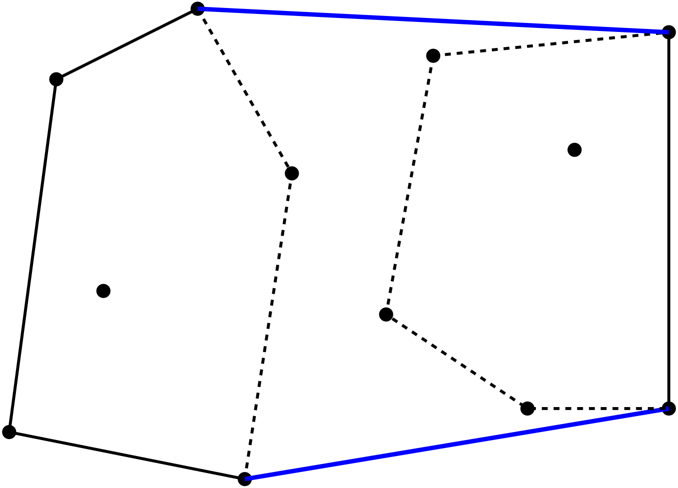
Example: Convex Hull

- $n = 12$ and $c = 2$



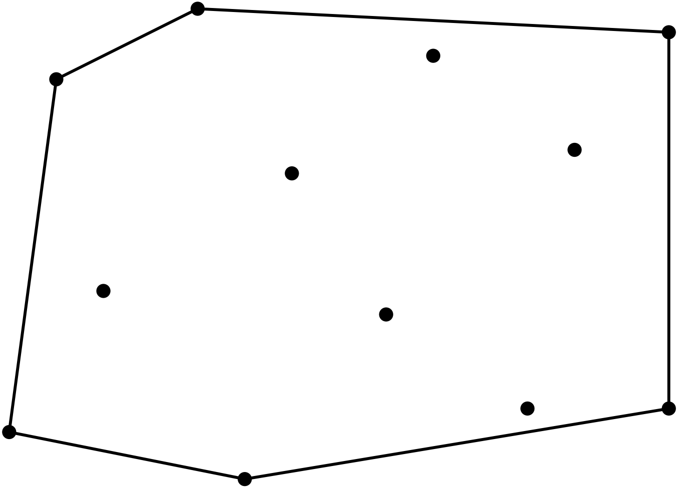
Example: Convex Hull

- $n = 12$ and $c = 2$



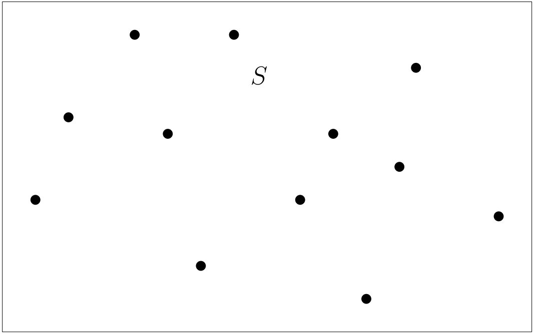
Example: Convex Hull

- $n = 12$ and $c = 2$

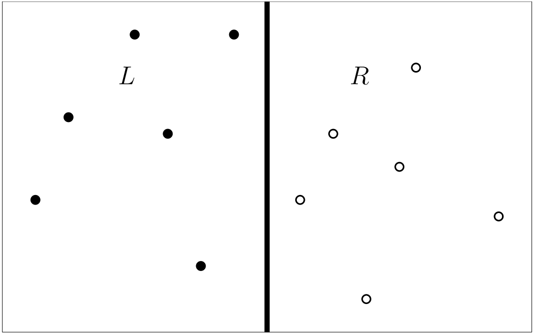


D & C for Voronoi Diagram

- Divide and Conquer to Compute $V(S)$
  1. Use a vertical line to partition S into L and R where $|L| \sim |R|$
     - If $|S|$ is a constant, directly compute $V(S)$ instead
  2. Recursively compute $V(L)$ and $V(R)$
  3. Merge $V(L)$ and $V(R)$ into $V(S)$

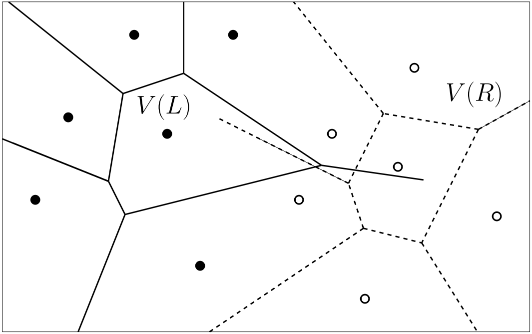


- Divide and Conquer to Compute $V(S)$
  1. Use a vertical line to partition S into L and R where $|L| \sim |R|$
     - If $|S|$ is a constant, directly compute $V(S)$ instead
  2. Recursively compute $V(L)$ and $V(R)$
  3. Merge $V(L)$ and $V(R)$ into $V(S)$



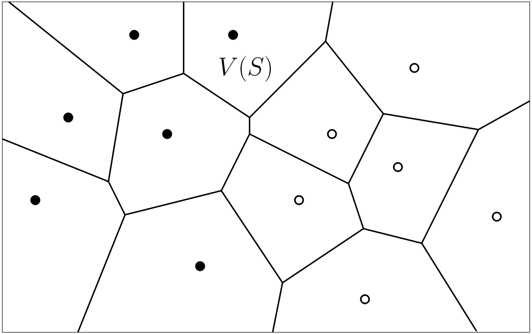
D & C for Voronoi Diagram

- Divide and Conquer to Compute $V(S)$
  1. Use a vertical line to partition S into L and R where $|L| \sim |R|$
     - If $|S|$ is a constant, directly compute $V(S)$ instead
  2. Recursively compute $V(L)$ and $V(R)$
  3. Merge $V(L)$ and $V(R)$ into $V(S)$



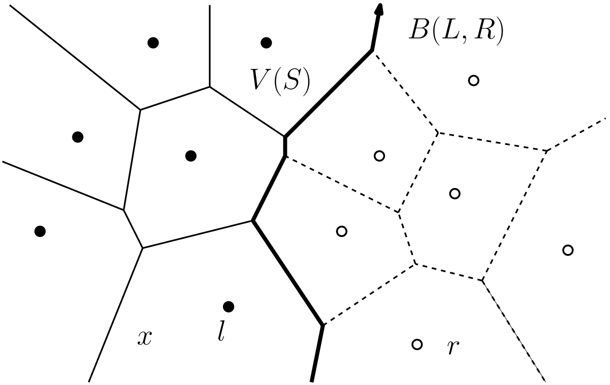
D & C for Voronoi Diagram

- Divide and Conquer to Compute $V(S)$
  1. Use a vertical line to partition S into L and R where $|L| \sim |R|$
     - If $|S|$ is a constant, directly compute $V(S)$ instead
  2. Recursively compute $V(L)$ and $V(R)$
  3. Merge $V(L)$ and $V(R)$ into $V(S)$



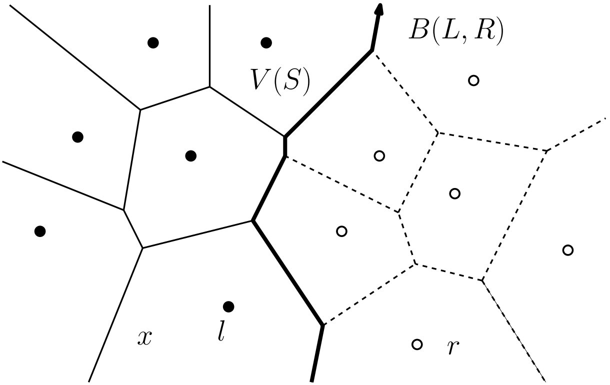
Merge Chain

- **Merge Chain** $B(L, R)$ consists of Voronoi edges between $V(l, S)$ and $V(r, S)$ where $l \in L$ and $r \in R$.



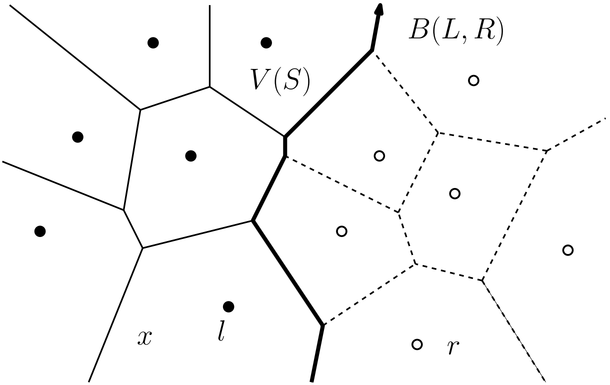
Merge Chain

- **Merge Chain** $B(L, R)$ consists of Voronoi edges between $V(l, S)$ and $V(r, S)$ where $l \in L$ and $r \in R$.
  - Voronoi edges between L's regions and R's regions



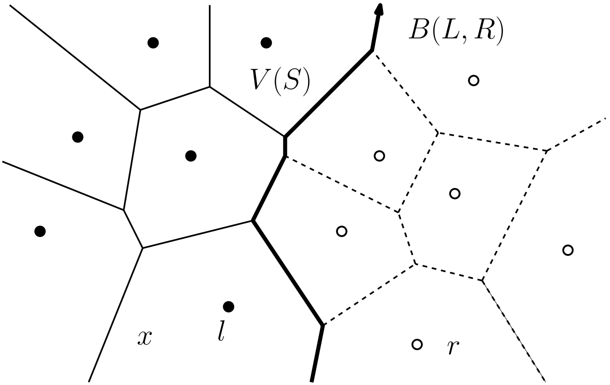
Merge Chain

- **Merge Chain** $B(L, R)$ consists of Voronoi edges between $V(l, S)$ and $V(r, S)$ where $l \in L$ and $r \in R$.
  - Voronoi edges between L's regions and R's regions
  - $d(x, L) = \min_{l \in L} d(x, l)$ (reps. $d(x, R)$)



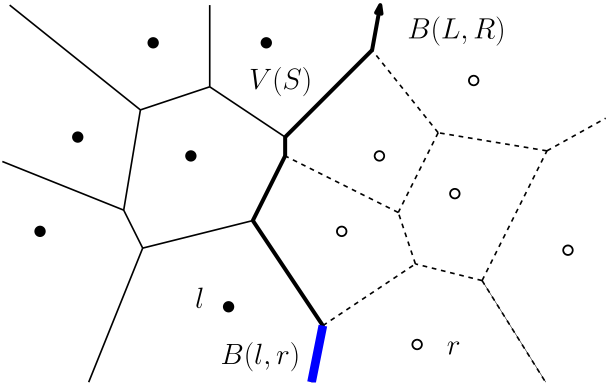
Merge Chain

- **Merge Chain** $B(L, R)$ consists of Voronoi edges between $V(l, S)$ and $V(r, S)$ where $l \in L$ and $r \in R$.
  - Voronoi edges between L's regions and R's regions
  - $d(x, L) = \min_{l \in L} d(x, l)$ (reps. $d(x, R)$)
  - $B(L, R) = \{x \in R^2 \mid d(x, L) = d(x, R)\}$



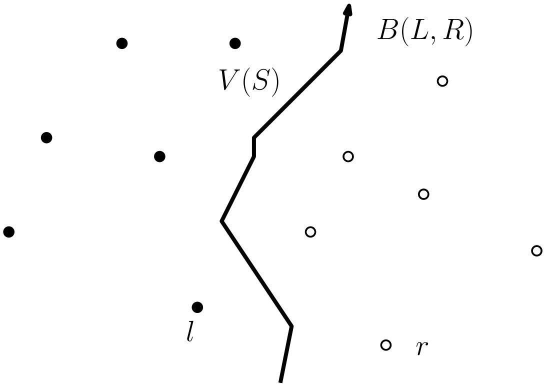
Merge Chain

- **Merge Chain** $B(L, R)$ consists of Voronoi edges between $V(l, S)$ and $V(r, S)$ where $l \in L$ and $r \in R$.
  - Voronoi edges between L's regions and R's regions
  - $d(x, L) = \min_{l \in L} d(x, l)$ (reps. $d(x, R)$)
  - $B(L, R) = \{x \in R^2 \mid d(x, L) = d(x, R)\}$
    - $\forall e \in B(L, R)$ belongs to $B(l, r)$, $l \in L$ and $r \in R$



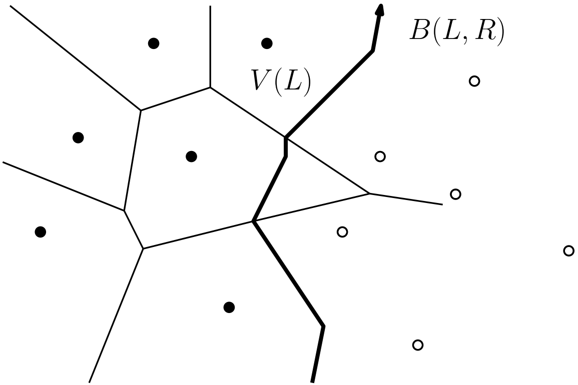
If We Have the Merge Chain...

- Compute $V(S)$
  - Remove the part of $V(L)$ right to $B(L, R)$
  - Remove the part of $V(R)$ left to $B(L, R)$
  - Glue the two remaining parts



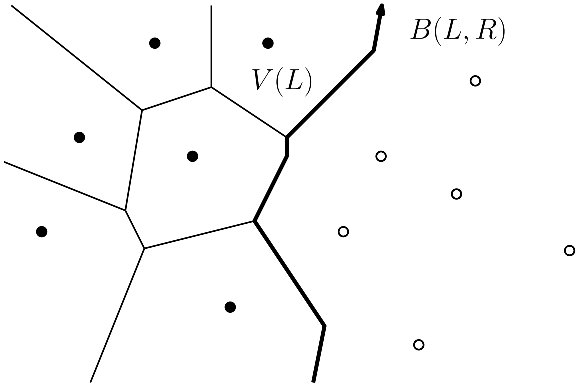
If We Have the Merge Chain...

- Compute $V(S)$
  - Remove the part of $V(L)$ right to $B(L, R)$
  - Remove the part of $V(R)$ left to $B(L, R)$
  - Glue the two remaining parts



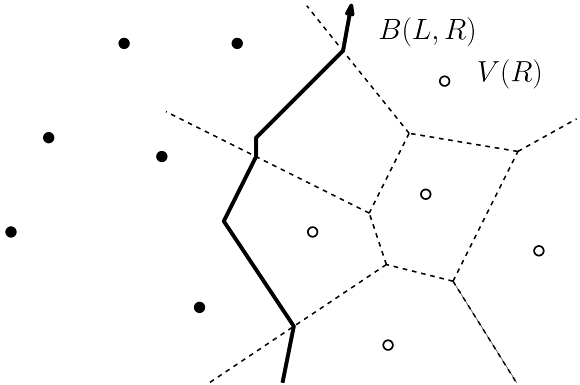
If We Have the Merge Chain...

- Compute $V(S)$
  - Remove the part of $V(L)$ right to $B(L, R)$
  - Remove the part of $V(R)$ left to $B(L, R)$
  - Glue the two remaining parts



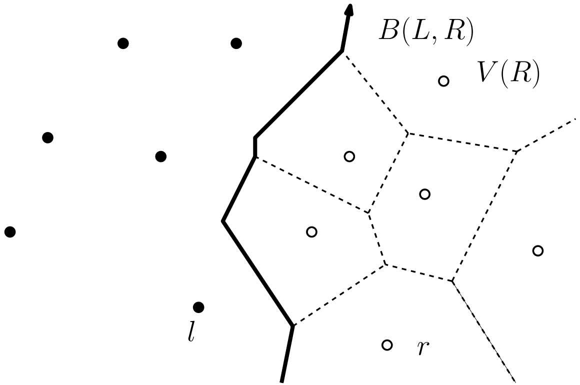
If We Have the Merge Chain...

- Compute $V(S)$
  - Remove the part of $V(L)$ right to $B(L, R)$
  - Remove the part of $V(R)$ left to $B(L, R)$
  - Glue the two remaining parts



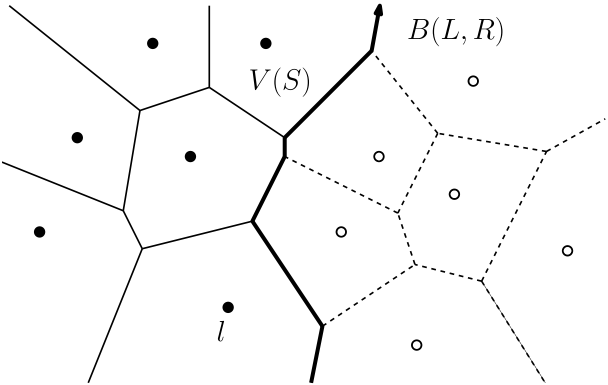
If We Have the Merge Chain...

- Compute $V(S)$
  - Remove the part of $V(L)$ right to $B(L, R)$
  - Remove the part of $V(R)$ left to $B(L, R)$
  - Glue the two remaining parts



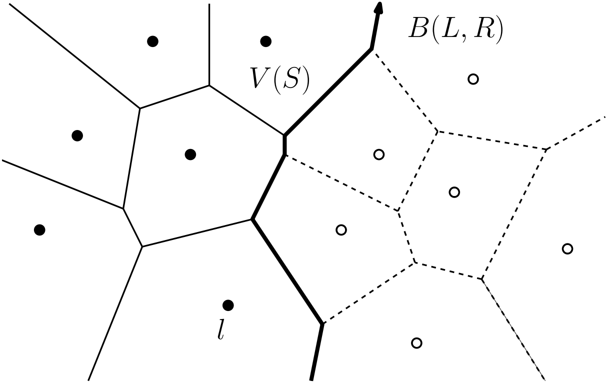
If We Have the Merge Chain...

- Compute $V(S)$
  - Remove the part of $V(L)$ right to $B(L, R)$
  - Remove the part of $V(R)$ left to $B(L, R)$
  - Glue the two remaining parts



If We Have the Merge Chain...

- Compute $V(S)$
  - Remove the part of $V(L)$ right to $B(L,R)$
  - Remove the part of $V(R)$ left to $B(L,R)$
  - Glue the two remaining parts
- $B(L,R)$ is a part of $V(S)$



$B(L, R)$ is y-monotone

Lemma

$B(L, R)$ is y-monotone

- Let b be an edge of $B(L, R)$.
- b belongs to $V(l, S)$ and $V(r, S)$, $l \in L$ and $r \in R$
- Let b be directed such that l in the left of b
- x-coordinate of $l <$ x-coordinate of r
  $\rightarrow$ b must be **upward**

$B(L, R)$ is y-monotone

Lemma

$B(L, R)$ is y-monotone

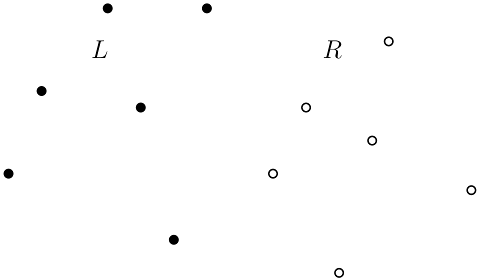
- Let b be an edge of $B(L, R)$.
- b belongs to $V(l, S)$ and $V(r, S)$, $l \in L$ and $r \in R$
- Let b be directed such that l in the left of b
- x-coordinate of $l < x$-coordinate of r
  $\rightarrow b$ must be **upward**

Computing $B(L, R)$

- Find the bottom edge e of $B(L, R)$ as a starting edge
  - e is unbounded and extends to $-\infty$
- Trace out $B(L, R)$ from e

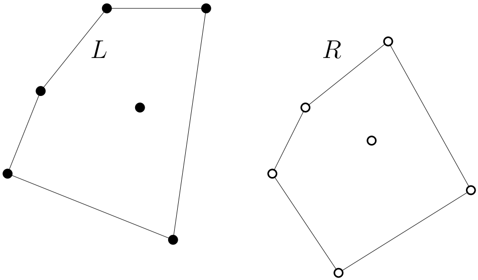
Finding a starting edge of $B(L, R)$

- An unbounded Voronoi edge corresponds to an edge of the convex hull $conv(S)$ of S



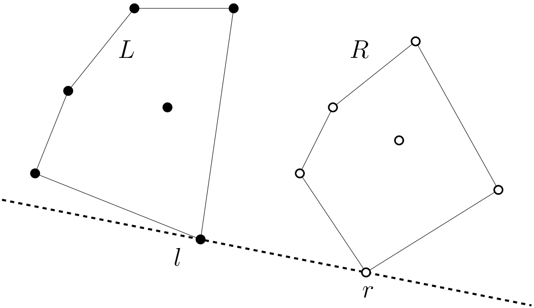
Finding a starting edge of $B(L, R)$

- An unbounded Voronoi edge corresponds to an edge of the convex hull $conv(S)$ of S
  - Build $conv(L)$ and $conv(R)$
    - $V(L)$ and $V(R)$ are known.



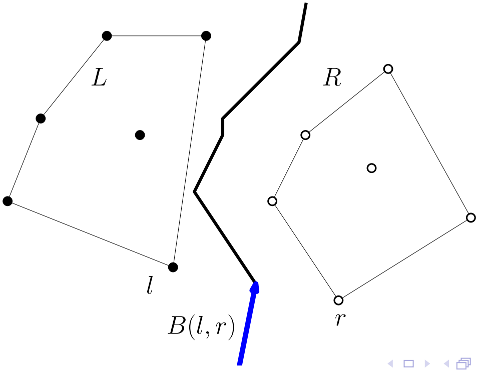
Finding a starting edge of $B(L, R)$

- An unbounded Voronoi edge corresponds to an edge of the convex hull $conv(S)$ of S
  - Build $conv(L)$ and $conv(R)$
    - $V(L)$ and $V(R)$ are known.
  - Compute a tangent edge $\overline{lr}$ between $conv(L)$ and $conv(R)$



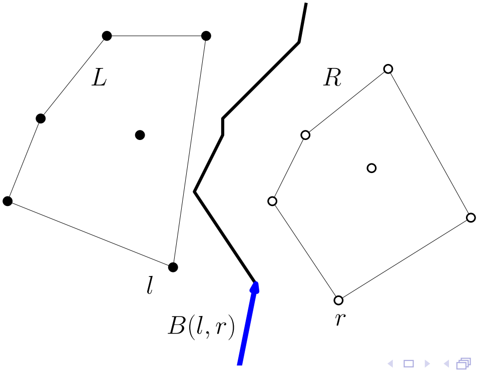
Finding a starting edge of $B(L, R)$

- An unbounded Voronoi edge corresponds to an edge of the convex hull $conv(S)$ of S
  - Build $conv(L)$ and $conv(R)$
    - $V(L)$ and $V(R)$ are known.
  - Compute a tangent edge $\overline{lr}$ between $conv(L)$ and $conv(R)$
  - $B(l, r)$ is the starting edge



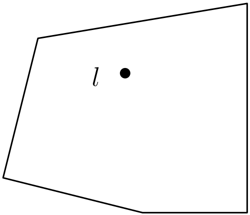
Finding a starting edge of $B(L, R)$

- An unbounded Voronoi edge corresponds to an edge of the convex hull $conv(S)$ of S
  - Build $conv(L)$ and $conv(R)$
    - $V(L)$ and $V(R)$ are known.
  - Compute a tangent edge $\overline{lr}$ between $conv(L)$ and $conv(R)$
  - $B(l, r)$ is the starting edge
- $O(|L| + |R|)$ time



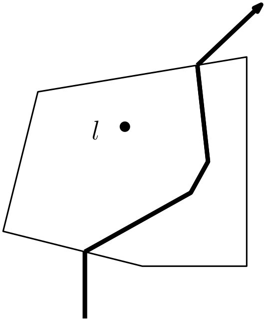
Trace $B(L, R)$ in $V(l, L)$ (1)

- $B(L, R) \cap V(l, L)$
  - $B(L, R)$ enters $V(l, L)$ at v along $B(l, r_1)$



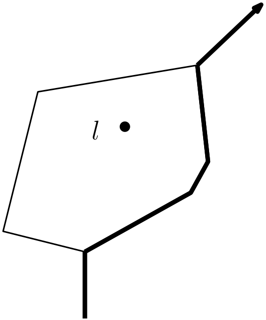
Trace $B(L,R)$ in $V(l,L)$ (1)

- $B(L,R) \cap V(l,L)$
  - $B(L,R)$ enters $V(l,L)$ at v along $B(l,r_1)$



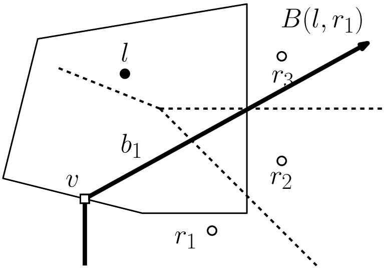
Trace $B(L, R)$ in $V(l, L)$ (1)

- $B(L, R) \cap V(l, L)$
  - $B(L, R)$ enters $V(l, L)$ at v along $B(l, r_1)$



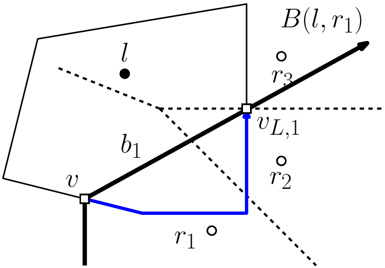
Trace $B(L, R)$ in $V(l, L)$ (1)

- $B(L, R) \cap V(l, L)$
  - $B(L, R)$ enters $V(l, L)$ at v along $B(l, r_1)$



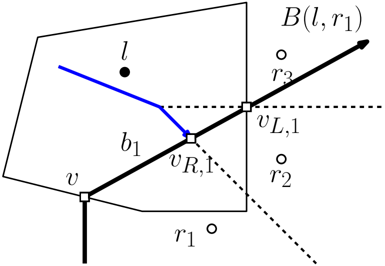
Trace $B(L, R)$ in $V(l, L)$ (1)

- $B(L, R) \cap V(l, L)$
  - $B(L, R)$ enters $V(l, L)$ at v along $B(l, r_1)$
  - Counterclockwise along $\partial V(l, L)$ from v to find $v_{L,1} \in B(l, r_1)$
    - After $v_{L,1}$, $V(l, L) \to V(l', L)$, and $B(l, r_1) \to B(l', r_1)$



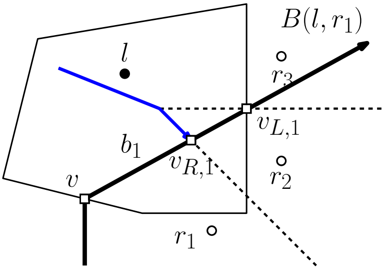
Trace $B(L, R)$ in $V(l, L)$ (1)

- $B(L, R) \cap V(l, L)$
  - $B(L, R)$ enters $V(l, L)$ at v along $B(l, r_1)$
  - Counterclockwise along $\partial V(l, L)$ from v to find $v_{L,1} \in B(l, r_1)$
    - After $v_{L,1}$, $V(l, L) \to V(l', L)$, and $B(l, r_1) \to B(l', r_1)$
  - Clockwise along $\partial V(r_1, R)$ to find $v_{R,1} \in B(l, r_1)$
    - After $v_{R,1}$, $V(r_1, R) \to V(r_2, R)$, and $B(l, r_1) \to B(l, r_2)$



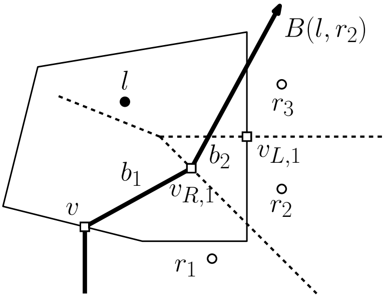
Trace $B(L, R)$ in $V(l, L)$ (1)

- $B(L, R) \cap V(l, L)$
  - $B(L, R)$ enters $V(l, L)$ at v along $B(l, r_1)$
  - Counterclockwise along $\partial V(l, L)$ from v to find $v_{L,1} \in B(l, r_1)$
    - After $v_{L,1}$, $V(l, L) \to V(l', L)$, and $B(l, r_1) \to B(l', r_1)$
  - Clockwise along $\partial V(r_1, R)$ to find $v_{R,1} \in B(l, r_1)$
    - After $v_{R,1}$, $V(r_1, R) \to V(r_2, R)$, and $B(l, r_1) \to B(l, r_2)$
  - $v_{R,1}$ earlier than $v_{L,1}$, and $B(l, r_1) \to B(l, r_2)$ at $v_{R,1}$



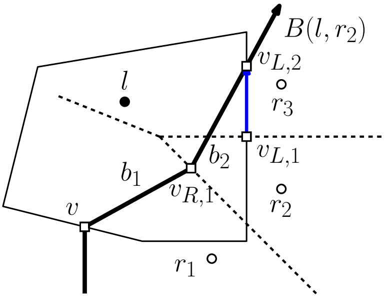
Trace $B(L, R)$ in $V(l, L)$ (2)

- $B(L, R) \cap V(l, L)$
  - $B(L, R)$ from $v_{R,1}$ along $B(l, r_2)$



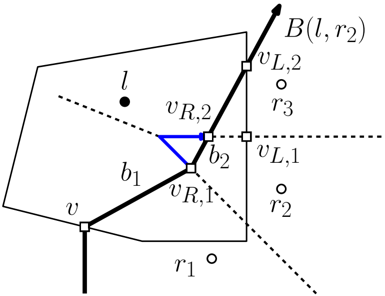
Trace $B(L,R)$ in $V(l,L)$ (2)

- $B(L,R) \cap V(l,L)$
  - $B(L,R)$ from $v_{R,1}$ along $B(l,r_2)$
  - Counterclockwise along $\partial V(l,L)$ from $v_{L,1}$ to find $v_{L,2} \in B(l,r_2)$
    - After $v_{L,2}$, $V(l,L) \to V(l',L)$, and $B(l,r_2) \to B(l',r_2)$



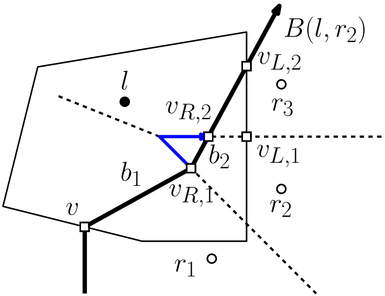
Trace $B(L,R)$ in $V(l,L)$ (2)

- $B(L,R) \cap V(l,L)$
  - $B(L,R)$ from $v_{R,1}$ along $B(l,r_2)$
  - Counterclockwise along $\partial V(l,L)$ from $v_{L,1}$ to find $v_{L,2} \in B(l,r_2)$
    - After $v_{L,2}$, $V(l,L) \rightarrow V(l',L)$, and $B(l,r_2) \rightarrow B(l',r_2)$
  - Clockwise along $\partial V(r_2,R)$ from $v_{R,1}$ to find $v_{R,2} \in B(l,r_2)$
    - After $v_{R,2}$, $V(r_2,R) \rightarrow V(r_3,R)$, and $B(l,r_2) \rightarrow B(l,r_3)$



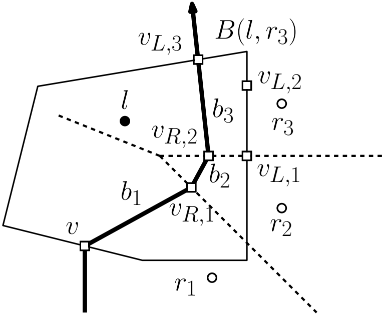
Trace $B(L, R)$ in $V(l, L)$ (2)

- $B(L, R) \cap V(l, L)$
  - $B(L, R)$ from $v_{R,1}$ along $B(l, r_2)$
  - Counterclockwise along $\partial V(l, L)$ from $v_{L,1}$ to find $v_{L,2} \in B(l, r_2)$
    - After $v_{L,2}$, $V(l, L) \to V(l', L)$, and $B(l, r_2) \to B(l', r_2)$
  - Clockwise along $\partial V(r_2, R)$ from $v_{R,1}$ to find $v_{R,2} \in B(l, r_2)$
    - After $v_{R,2}$, $V(r_2, R) \to V(r_3, R)$, and $B(l, r_2) \to B(l, r_3)$
  - $v_{R,2}$ earlier than $v_{L,2}$, and $B(l, r_2) \to B(l, r_3)$ at $v_{R,2}$



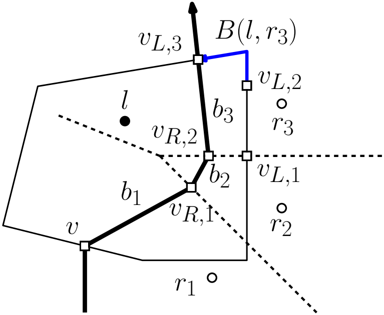
Trace $B(L, R)$ in $V(l, L)$ (3)

- $B(L, R) \cap V(l, L)$
  - $B(L, R)$ from $v_{R,2}$ along $B(l, r_3)$



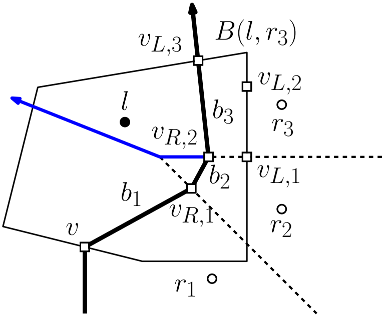
Trace $B(L, R)$ in $V(l, L)$ (3)

- $B(L, R) \cap V(l, L)$
  - $B(L, R)$ from $v_{R,2}$ along $B(l, r_3)$
  - Counterclockwise along $\partial V(l, L)$ from $v_{L,2}$ to find $v_{L,3} \in B(l, r_3)$
    - After $v_{L,3}$, $V(l, L) \to V(l'', L)$, and $B(l, r_3) \to B(l'', r_3)$



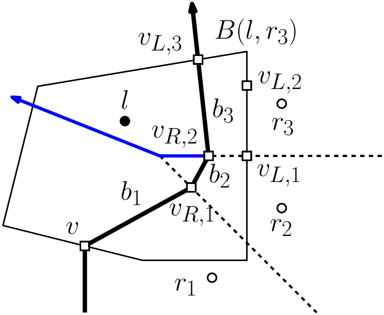
Trace $B(L, R)$ in $V(l, L)$ (3)

- $B(L, R) \cap V(l, L)$
  - $B(L, R)$ from $v_{R,2}$ along $B(l, r_3)$
  - Counterclockwise along $\partial V(l, L)$ from $v_{L,2}$ to find $v_{L,3} \in B(l, r_3)$
    - After $v_{L,3}$, $V(l, L) \to V(l'', L)$, and $B(l, r_3) \to B(l'', r_3)$
  - Clockwise along $\partial V(r_3, R)$ from $v_{R,2}$ to find $v_{R,3} \in B(l, r_3)$
    - No $v_{R,3}$



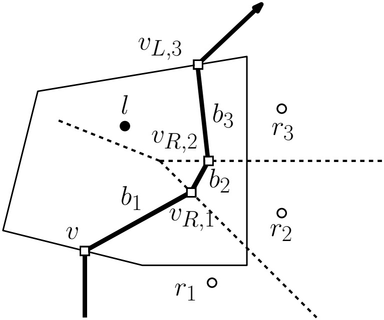
Trace $B(L, R)$ in $V(l, L)$ (3)

- $B(L, R) \cap V(l, L)$
  - $B(L, R)$ from $v_{R,2}$ along $B(l, r_3)$
  - Counterclockwise along $\partial V(l, L)$ from $v_{L,2}$ to find $v_{L,3} \in B(l, r_3)$
    - After $v_{L,3}$, $V(l, L) \to V(l'', L)$, and $B(l, r_3) \to B(l'', r_3)$
  - Clockwise along $\partial V(r_3, R)$ from $v_{R,2}$ to find $v_{R,3} \in B(l, r_3)$
    - No $v_{R,3}$
  - $B(l, r_3) \to B(l'', r_3)$ at $v_{L,3}$



Trace $B(L, R)$ in $V(l, L)$ (3)

- $B(L, R) \cap V(l, L)$
  - $B(L, R)$ from $v_{R,2}$ along $B(l, r_3)$
  - Counterclockwise along $\partial V(l, L)$ from $v_{L,2}$ to find $v_{L,3} \in B(l, r_3)$
    - After $v_{L,3}$, $V(l, L) \to V(l'', L)$, and $B(l, r_3) \to B(l'', r_3)$
  - Clockwise along $\partial V(r_3, R)$ from $v_{R,2}$ to find $v_{R,3} \in B(l, r_3)$
    - No $v_{R,3}$
  - $B(l, r_3) \to B(l'', r_3)$ at $v_{L,3}$



Time to Compute $B(L, R)$

Lemma

Computing $B(L, R)$ takes $O(|L| + |R|) = O(|S|)$ time

- Finding the starting edge takes $O(|L| + |R|)$ time
- Traversing $V(L)$ takes $O(|V(L)|)$ time ($V(R) \rightarrow O(|V(R)|)$)
  - Enter each $V(l, L)$ and its boundary at most once
- There are $O(|B(L, R)|)$ intersections
  - Each edge of $B(L, R)$ makes two intersections.
- $O(|L| + |R|) + O(|V(L)|) + O(|V(R)|) + O(|B(L, R)|)$ $= O(|L| + |R| + |L| + |L| + |S|) = O(|S|)$
  - $O(|V(L)|) = O(|L|)$, $O(|V(R)|) = O(|R|)$, and $O(|B(L, R)|) = O(|S|)$

Time Complexity

Theorem

The divide-and-conquer algorithm computes $V(S)$ in $O(n \log n)$ time

- Sorting takes $O(n \log n)$ time (only do once)
- A sub-instance $S' \subset S$ takes $O(|S'|)$ time
  - Partitioning S' into L' and R' takes $O(|S'|)$ time
  - Computing $B(L', R')$ takes $O(|S'|)$ time
- The i^{th}-level takes $O(2^i) \times O(n/2^i) = O(n)$ time
  - $O(2^i)$ sub-instances each with $O(n/2^i)$ sites
- $O(\log n)$ levels

Time Complexity

Theorem

The divide-and-conquer algorithm computes $V(S)$ in $O(n \log n)$ time

- Sorting takes $O(n \log n)$ time (only do once)
- A sub-instance $S' \subset S$ takes $O(|S'|)$ time
  - Partitioning S' into L' and R' takes $O(|S'|)$ time
  - Computing $B(L', R')$ takes $O(|S'|)$ time
- The i^{th}-level takes $O(2^i) \times O(n/2^i) = O(n)$ time
  - $O(2^i)$ sub-instances each with $O(n/2^i)$ sites
- $O(\log n)$ levels
- $T(n) = O(n) + 2 \cdot T(n/2) => T(n) = O(n \log n)$