Voronoi diagram

Given points \( P = \{p_1, \ldots, p_n\} \) in the plane, the Voronoi region of \( p_i \) is

\[
V(p_i) = \{ x \in \mathbb{R}^2 : d(x, p_i) \leq d(x, p_j) \quad \forall j \neq i \}
\]

\( p_i \) is called a site.

The Voronoi diagram \( V(P) \) consists of all the Voronoi regions.

To see that the Voronoi diagram consists of straight line segments:

\[
H(p_i, p_j) = \text{points closer to } p_i \text{ than } p_j
\]

\[
= \text{half-plane}
\]

\[
V(p_i) = \bigcap_{j \neq i} H(p_i, p_j)
\]
RECALL
Terminology

Voronoi vertex

site

Voronoi edge

\( \text{Pi} \)
**RECALL**

**Delaunay triangulation**

The Voronoi diagram can be captured by a purely combinatorial structure (versus computing coordinates of Voronoi vertices).

Given points \( P = \{p_1, \ldots, p_n\} \) in the plane, the *Delaunay triangulation* \( \mathcal{D}(P) \) is a graph with vertices \( p_1, \ldots, p_n \) and edge \( (p_i, p_j) \) iff \( V(p_i) \) and \( V(p_j) \) share an edge.

This is the planar dual of \( \mathcal{V}(P) \).
Recall properties of Voronoi diagram, Delaunay triangulation

- Voronoi regions are convex polygons
- Vor. vertices have deg. 3 (assuming no 4 sites on circle)
- Vor. region unbounded \( \Leftrightarrow \) site on CH
- Delaunay triangulation is a triangulation.
- \((u,v)\) is Delaunay edge \( \Leftrightarrow \) \exists empty circle through \(u\) and \(v\)
- \(u,v,w\) is a triangular face \( \Leftrightarrow \) \exists empty circle through \(u,v,w\).
- boundary of \(D(P)\) = convex hull of \(P\).
Application of Delaunay triangulations: finding all nearest neighbours

Given $n$ points in the plane find, for each point, its nearest neighbour — gives *nearest neighbour graph*, a directed graph of out-degree 1.

Many applications, e.g.

in statistical analysis:
find hierarchical clusters using nearest neighbour chain algorithm
the nearest neighbour graph
can also look at k nearest neighbours — use k-th order Voronoi diagrams (later)
Application of Delaunay triangulations: finding min spanning trees (MST)

Given points \( p_1, \ldots, p_n \) in the plane, find the Euclidean minimum spanning tree

\[ = \text{tree with vertex set } p_1, \ldots, p_n \text{ of minimum total length} \]

There are good algorithms to find the min weight spanning tree in any edge-weighted graph.
But our graph has \( O(n^2) \) edges.

Lemma. The minimum spanning tree is a subgraph of the Delaunay triangulation.

Then we can run the graph MST algorithm on the Delaunay triangulation to get an
\( O(n \log n) \) algorithm.

Proof of Lemma.
Relative Neighbourhood and Gabriel graphs
Application of Delaunay triangulations: finding min spanning trees (MST)

Given $n$ points in a convex boundary polygon $B$, find the largest empty circle with center in $B$.

E.g. locate a new store location among existing stores.
   - locate a nuclear waste dump among cities.

Solution
Connections between Voronoi diagram / Delaunay triangulation and Convex Hull

Given \( p_1, \ldots, p_n \in \mathbb{R}^2 \) project them up onto parabola \( z = x^2 + y^2 \)
\[
p = (x_p, y_p) \quad \rightarrow \quad \hat{p} = (x_p, y_p, x_p^2 + y_p^2)
\]

Theorem. The lower convex hull of \( \hat{p}_1, \ldots, \hat{p}_n \), projected back to the plane, is the Delaunay triangulation of \( p_1, \ldots, p_n \).
**Figure 1.11.** Points $a$, $b$, $c$ lie on the dashed circle in the $x_1x_2$-plane and $d$ lies inside that circle. The dotted curve is the intersection of the paraboloid with the plane that passes through $\tilde{a}$, $\tilde{b}$, $\tilde{c}$. It is an ellipse whose projection is the dashed circle.
Algorithms to compute Voronoi diagrams / Delaunay triangulations

- note that we can get either one from the other in $O(n)$ time.

- we can compute the Delaunay triangulation in $O(n \log n)$ time using a 3D convex hull algorithm.

- first $O(n \log n)$ algorithm to compute Voronoi diagram was divide and conquer, Shamos and Hoey, 1975. The merge step is complicated.

- Steve Fortune, ’87, gave a sweepline algorithm for Voronoi diagram

- randomized incremental algorithm to compute the Delaunay triangulation — next lecture.
Fortune’s sweepline algorithm for Voronoi diagram

the difficulty with a sweepline approach:

V(p) starts before we reach p

Solution
intermediate configuration of Fortune’s algorithm

Voronoi edges

"beachfront" of parabolic sections

https://www.youtube.com/watch?v=rvmREoyL2F0
update events for Fortune’s algorithm

reach a new point

a new parabolic segment appears

a parabolic section vanishes. Our “event list” must include $y=k$
Another way to visualize Fortune’s algorithm

the Voronoi diagram can be viewed as the projection of the upper envelope of cones

and Fortune’s algorithm sweeps a plane $\pi$ across those cones
Other versions of Voronoi diagrams

- the sites may be more general than points, e.g. line segments, polygons, etc.

- higher dimensions

- farthest point Voronoi diagrams
Other versions of Voronoi diagrams

- weighted Voronoi diagrams

- Voronoi diagrams for other distance metrics