Lecture 12:
Voronoi Diagrams,
Part 3
Outline for Today

● Homework 4 Posted!
● Friday Feb 25th: Quiz 1, Logistics
● Last Time: More Voronoi Diagrams
● Higher-Order vs Higher Dimension Voronoi Diagrams
● Centroidal Voronoi Diagram
● K-Means Clustering
● Application: Architectural Geometry
● More Spatial Query & Search Problems / Applications
● Reducing Other Problems to the Voronoi Diagram
● Future Topic (after Quiz 1): Delaunay Triangulation
Quiz 1

- In class, Friday Feb 25th, 2-3:50pm
- On paper, will involve simple sketching, you are welcome (but not required) to bring colored pencils/markers/crayons/etc.
- If you cannot attend class in person, contact me at least 24 hours in advance for alternate arrangements.

- Volunteer to pick up the quizzes from Shannon or Shianne Friday morning/early afternoon & slide under my office door in Lally after the quiz?
Outline for Today

- Homework 4 Posted!
- Friday Feb 25th: Quiz 1, Logistics
- Last Time: More Voronoi Diagrams
- Higher-Order vs Higher Dimension Voronoi Diagrams
- Centroidal Voronoi Diagram
- K-Means Clustering
- Application: Architectural Geometry
- More Spatial Query & Search Problems / Applications
- Reducing Other Problems to the Voronoi Diagram
- Future Topic (after Quiz 1): Delaunay Triangulation
Voronoi Diagram of Line Segments

- Points equidistant between two points form a line.
- Points equidistant between a point and a line form a parabola.
- Points equidistant between two lines form a line.
Sweep Line: More Complicated Beach Front

- Fortunately, the complexity (# of segments) is still $O(n)$ in the size of the input – now line segments instead of just points!
Application: Robotics & Motion Planning

- Step 1: Project robot center to closest Voronoi edge.
- Step 2: Remove Voronoi edges from diagram graph where smallest distance to segment < radius.
- Step 3: Search the remaining graph for a connected path from start to end.
The intersection of these half-spaces is the Voronoi Cell for A – all points that choose A as their closest Voronoi site.
Definition: Farthest Point Voronoi Cell

- The intersection of these half-spaces is the Voronoi Cell for A – all points that choose A as their closest Voronoi site.

- The intersection of the opposite half-space is the Farthest Point Voronoi Cell - all points that indicate that A is their furthest Voronoi site.
Farthest-Point Voronoi Diagram

- Observation: Only sites on the convex hull will have a cell in the farthest point diagram.

- Observation: All farthest-point cells are unbounded.

- Observation: The diagram is a tree – no cycles!

*If there were a cycle, that would mean we had a bounded cell.*
Finding the Smallest-Width Annulus

- Easy to compute once we know the center (it is the center of both the inner & outer circle)
- *What points might be the center? Any point on the plane?*

*Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 7*
Finding the Smallest-Width Annulus

- Easy to compute once we know the center (it is the center of both the inner & outer circle)
- *What points might be the center?*
  It must be:
  - A vertex of the Voronoi Diagram (equally close to 3 sites) OR
  - A vertex of the Farthest Point Voronoi Diagram (equally far from 3 sites) OR
  - An intersection of the Voronoi Diagram and Farthest Point Voronoi Diagram (equally close to 2 sites AND equally far from 2 sites)
Finding the Smallest-Width Annulus

- Easy to compute once we know the center (it is the center of both the inner & outer circle)
- **What points might be the center?**
  It must be:
  - A vertex of the Voronoi Diagram (equally close to 3 sites) OR
  - A vertex of the Farthest Point Voronoi Diagram (equally far from 3 sites) OR
  - An intersection of the Voronoi Diagram and Farthest Point Voronoi Diagram (equally close to 2 sites AND equally far from 2 sites)
Finding the Smallest-Width Annulus

- Easy to compute once we know the center (it is the center of both the inner & outer circle)
- **What points might be the center?**
  It must be:
  - A vertex of the Voronoi Diagram (equally close to 3 sites) OR
  - A vertex of the Farthest Point Voronoi Diagram (equally far from 3 sites) OR
  - An intersection of the Voronoi Diagram and Farthest Point Voronoi Diagram (equally close to 2 sites AND equally far from 2 sites)
Finding the Smallest-Width Annulus

- Easy to compute once we know the center (it is the center of both the inner & outer circle)
- **What points might be the center?**
  It must be:
  - A vertex of the Voronoi Diagram (equally close to 3 sites) OR
  - A vertex of the Farthest Point Voronoi Diagram (equally far from 3 sites) OR
  - An intersection of the Voronoi Diagram and Farthest Point Voronoi Diagram (equally close to 2 sites AND equally far from 2 sites)
Finding the Smallest-Width Annulus

- Easy to compute once we know the center (it is the center of both the inner & outer circle)
- **What points might be the center?**
  
  It must be:
  
  - A vertex of the Voronoi Diagram (equally close to 3 sites) OR
  - A vertex of the Farthest Point Voronoi Diagram (equally far from 3 sites) OR
  - An intersection of the Voronoi Diagram and Farthest Point Voronoi Diagram (equally close to 2 sites AND equally far from 2 sites)

Brute force check of $O(n) = a$ FINITE number of possible center positions

*Computational Geometry Algorithms and Applications, de Berg, Cheong, van Kreveld and Overmars, Chapter 7*
Outline for Today

- Homework 4 Posted!
- Friday Feb 25th: Quiz 1, Logistics
- Last Time: More Voronoi Diagrams
- Higher-Order vs Higher Dimension Voronoi Diagrams
- Centroidal Voronoi Diagram
- K-Means Clustering
- Application: Architectural Geometry
- More Spatial Query & Search Problems / Applications
- Reducing Other Problems to the Voronoi Diagram
- Future Topic (after Quiz 1): Delaunay Triangulation
Higher-Order Voronoi / k-Closest Sites

- For example, k = 2…
- Subdivide the plane into regions that have the same closest and second closest sites
Voronoi Diagram in 3D or Higher Dimension

- Not the same as “Higher-Order Voronoi Diagram”
- Well defined in higher dimensions, but hard to visualize & debug!
- Each Voronoi cell is convex!


“Simulation and Optimization of Porous Bone-Like Microstructures with Specific Mechanical Properties”, Wit, Wronski, & Tarasiuk, 2019
Outline for Today

- Homework 4 Posted!
- Friday Feb 25th: Quiz 1, Logistics
- Last Time: More Voronoi Diagrams
- Higher-Order vs Higher Dimension Voronoi Diagrams
- Centroidal Voronoi Diagram
- K-Means Clustering
- Application: Architectural Geometry
- More Spatial Query & Search Problems / Applications
- Reducing Other Problems to the Voronoi Diagram
- Future Topic (after Quiz 1): Delaunay Triangulation
Centroidal Voronoi Diagram

- What if we could place all of the grocery stores?
- Where should we place the grocery stores so that they are centrally located for all of their customers?

https://en.wikipedia.org/wiki/Centroidal_Voronoi_tessellation
Centroidal Voronoi Diagram

- What if we could place all of the grocery stores?
- Where should we place the grocery stores so that they are centrally located for all of their customers?
- But if you change the position of the store, the closest store will change for some customers…

- Points are at the center of mass of their cell
- Constructed using k-means clustering /
  Lloyd’s algorithm - an iterative relaxation algorithm

- Note: May be multiple solutions!

https://en.wikipedia.org/wiki/Centroidal_Voronoi_tessellation
Outline for Today

- Homework 4 Posted!
- Friday Feb 25th: Quiz 1, Logistics
- Last Time: More Voronoi Diagrams
- Higher-Order vs Higher Dimension Voronoi Diagrams
- Centroidal Voronoi Diagram
- **K-Means Clustering**
- Application: Architectural Geometry
- More Spatial Query & Search Problems / Applications
- Reducing Other Problems to the Voronoi Diagram
- Future Topic (after Quiz 1): Delaunay Triangulation
K-Means Clustering

- Works quite well, when the data can be meaningfully classified (and we know how many clusters to use).
- With dense data, output is visually similar to Voronoi diagram (k-Means chooses the data points that define the cells)


"Efficient K-Means Clustering using JIT" Yi Cao
K-Means Clustering

For a set of 2D/3D/nD points:

- Choose $k$, # of clusters (maybe an “oracle” tells us...)
- Select $k$ points from your data at random as initial team representatives
- Every other point determines which team representative it is closest to and joins that team
- The team averages the positions of all members, this is the team’s new representative
- Repeat $x$ times or until change $<$ threshold

Same/Similar to: Lloyd's Algorithm
Outline for Today

- Homework 4 Posted!
- Friday Feb 25th: Quiz 1, Logistics
- Last Time: More Voronoi Diagrams
- Higher-Order vs Higher Dimension Voronoi Diagrams
- Centroidal Voronoi Diagram
- K-Means Clustering
- **Application: Architectural Geometry**
- More Spatial Query & Search Problems / Applications
- Reducing Other Problems to the Voronoi Diagram
- Future Topic (after Quiz 1): Delaunay Triangulation
The Problem: Mesh this curved surface so that it can be constructed from glass for a rooftop greenhouse.

Mark Goulthorpe, MIT, dECOi
Great Court at the British Museum
London, England

Chadstone Shopping Center
Melbourne, Australia
“Voronoi Surfaces”

Unfortunately, the cell vertices are rarely planar!

Mike Powell, MIT
Studio Project Fall 2004
Voronoi Diagram on a 3D Surface

We're no longer using Euclidean Distance!
“Saddle Surface” → Non-Convex Facets

“bowtie” shapes
Fabrication
Additional work necessary to meet constraints of glass construction

“Voronoi Grid-Shell Structures”
Pietroni, Tonelli, Puppo, Froli, Scopigno, & Cignoni, 2014
“Geometric Modeling with Conical Meshes and Developable Surfaces”
Liu, Pottmann, Wallner, Yang & Wang, SIGGRAPH 2006
Voronoi Diagram in Nature

https://blogs.scientificamerican.com/observations/voronoi-tessellations-and-scutoids-are-everywhere/
Cellular Textures


Outline for Today

● Homework 4 Posted!
● Friday Feb 25th: Quiz 1, Logistics
● Last Time: More Voronoi Diagrams
● Higher-Order vs Higher Dimension Voronoi Diagrams
● Centroidal Voronoi Diagram
● K-Means Clustering
● Application: Architectural Geometry
● More Spatial Query & Search Problems / Applications
● Reducing Other Problems to the Voronoi Diagram
● Future Topic (after Quiz 1): Delaunay Triangulation
Problems that Reduce to Voronoi Diagram

- We can compute the Voronoi Diagram of \( n \) points in \( O(n \log n) \) time and \( O(n) \) space.
- These other problems can be computed in \( O(n) \) additional time if given the Voronoi Diagram.
- Therefore they are also \( O(n \log n) \) time and \( O(n) \) space.

*Computational Geometry: An Introduction, Preparata & Shamos, Figure 5.30*
Problem: All Nearest Neighbors

- Connect every point to its nearest point with a directed edge.
- Some points form a reciprocal pair.
Problem: Closest Pair

- Which two points are the closest?
- Applications - Collision Detection & Air Traffic Control
- Which two objects have soonest potential for collision?
Problem: Closest Pair

- Which two points are the closest?
- Applications - Collision Detection & Air Traffic Control
- Which two objects have soonest potential for collision?
- Linear loop over all edges in the All Nearest Neighbors solution to find the shortest edge
- Will be a reciprocal pair
Problem: Uniqueness

- Given $n$ numbers, decide if any two are unique.
Problem: Uniqueness

- Given \( n \) numbers, decide if any two are unique.

- Linear loop over all edges in the All Nearest Neighbors solution to if any edges are length zero.
Problem: Euclidean Minimum Spanning Tree

- Given $n$ points
- Draw $n-1$ edges to create a tree, connecting all points without creating any cycles.
- Pick edges to minimize the sum of their lengths.

**Application:** Minimize cost of physical telephone lines

![A minimum spanning tree on a planar point set.](image)

*Computational Geometry: An Introduction, Preparata & Shamos, Figure 5.2*
Problem: Minimum Spanning Tree

- General (non-Euclidean) MST from Graph Theory
- Each edge has a weight, not necessarily the Euclidean distance between two points
- Worst case, may have $m = n^2$ edges to consider
- Runtime $O(m \log n)$ $O(n^2 \log n)$ worst case

Figure 5.2 A minimum spanning tree on a planar point set.

Computational Geometry: An Introduction, Preparata & Shamos, Figure 5.2
Problem: (Euclidean) Steiner Tree

- If allowed to add additional points – so-called Steiner Points
- Minimize sum of Euclidean distance edge lengths
- *Computing the Steiner Tree is NP Complete / NP hard!*

Figure 5.3  A Steiner Tree (b) may have smaller total length than the MST (a).
Problems that Reduce to Voronoi Diagram

- We can compute the Voronoi Diagram of $n$ points in $O(n \log n)$ time and $O(n)$ space.
- These other problems can be computed in $O(n)$ additional time if given the Voronoi Diagram.
- Therefore they are also $O(n \log n)$ time and $O(n)$ space.
Reduce Convex Hull to Voronoi Diagram

- Theorem: Voronoi polygon $V_i$ is unbounded if and only if Voronoi site $i$ is on the convex hull of all sites. (proved in Preparata & Shamos)

Figure 5.31  Construction of the convex hull from the Voronoi diagram.
Reduce Convex Hull to Voronoi Diagram

- Theorem: Voronoi polygon $V_i$ is unbounded if and only if Voronoi site $i$ is on the convex hull of all sites. (proved in Preparata & Shamos)

- O(n) to convert Voronoi Diagram to Convex Hull:
  - Start with any unbounded cell
  - Walk edges clockwise to find adjacent unbounded cell
  - Voronoi sites will trace convex hull in counter-clockwise order

Figure 5.31 Construction of the convex hull from the Voronoi diagram.

*Computational Geometry: An Introduction, Preparata & Shamos, Figure 5.31*
Reduce All Nearest Neighbors to Voronoi Diagram

- For $n$ Voronoi Sites
- By Euler’s formula: $F + V = E + 2$
  - # of Voronoi edges $\leq 3n - 6$
- Theorem: Every nearest neighbor in the set of Voronoi sites defines an edge of a Voronoi polygon.
  (proved in Preparata & Shamos)

- $O(n)$ to convert Voronoi Diagram to All Nearest Neighbors
  - For every Voronoi polygon, loop over all Voronoi edges, & select adjacent site that is closest
  - Every Voronoi edge will be considered twice
Reduce All Nearest Neighbors to Voronoi Diagram

- For \( n \) Voronoi Sites
- By Euler’s formula: \( F + V = E + 2 \)
  - \# of Voronoi edges \( \leq 3n-6 \)
- Theorem: Every nearest neighbor in the set of Voronoi sites defines an edge of a Voronoi polygon.
  (proved in Preparata & Shamos)

*Computational Geometry: An Introduction, Preparata & Shamos, Figure 5.1*
Problems that Reduce to Voronoi Diagram

- We can compute the Voronoi Diagram of $n$ points in $O(n \log n)$ time and $O(n)$ space.
- These other problems can be computed in $O(n)$ additional time if given the Voronoi Diagram.
- Therefore they are also $O(n \log n)$ time and $O(n)$ space.

*Computational Geometry: An Introduction, Preparata & Shamos, Figure 5.30*
Reduce EMST to Voronoi Diagram

- All Nearest Neighbors will have one or more cycles reciprocal pair(s)
  - Remove one edge from each cycle
- All Nearest Neighbors may be disconnected
  - A forest of trees

Core challenge: Find and add shortest edge between disconnected trees in the forest…
Kruskal’s and Prim’s Algorithm for MST

Did you cover this in Introduction to Algorithms and/or Graph Theory?

- **Kruskal’s** - \(O(E \log E)\)
  - maintain a set of trees
  - find the shortest edge that merges two trees
  - repeat until there is only a single tree
- **Prim’s** - \(O(E + V \log V)\)
  - maintain one tree, and all unconnected vertices
  - find the shortest edge from the tree to an unconnected vertex
  - repeat until there are no unconnected vertices

For a general (non-Euclidean) MST, we may have to consider \(O(n^2)\) edges worst case. For the Euclidean MST, we only need to consider the \(O(n)\) Voronoi/Delaunay edges.
Problems that Reduce to Voronoi Diagram

- We can compute the Voronoi Diagram of \( n \) points in \( O(n \log n) \) time and \( O(n) \) space.
- These other problems can be computed in \( O(n) \) additional time if given the Voronoi Diagram.
- Therefore they are also \( O(n \log n) \) time and \( O(n) \) space.

*Computational Geometry: An Introduction, Preparata & Shamos, Figure 5.30*
Outline for Today

● Homework 4 Posted!
● Friday Feb 25th: Quiz 1, Logistics
● Last Time: More Voronoi Diagrams
● Higher-Order vs Higher Dimension Voronoi Diagrams
● Centroidal Voronoi Diagram
● K-Means Clustering
● Application: Architectural Geometry
● More Spatial Query & Search Problems / Applications
● Reducing Other Problems to the Voronoi Diagram
● Future Topic (after Quiz 1): Delaunay Triangulation
Next Time: Delaunay Triangulation!

- The Voronoi Diagram (VD) is the dual of the Delaunay Triangulation (DT)
- Every Voronoi Site is a face in Voronoi Diagram and a vertex in the DT
- Every Voronoi Edge is an edge in the DT
- Every Voronoi Vertex is a triangle in the DT

*Computational Geometry: An Introduction,* Preparata & Shamos, Figure 5.21