Voronoi Diagrams

Fortune's Algorithm and Applications

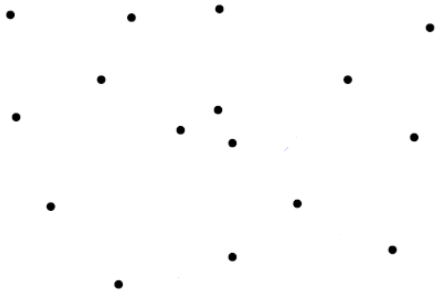
Kevin Wittmer
Mentor: Rob Maschal

DRP Summer 2016

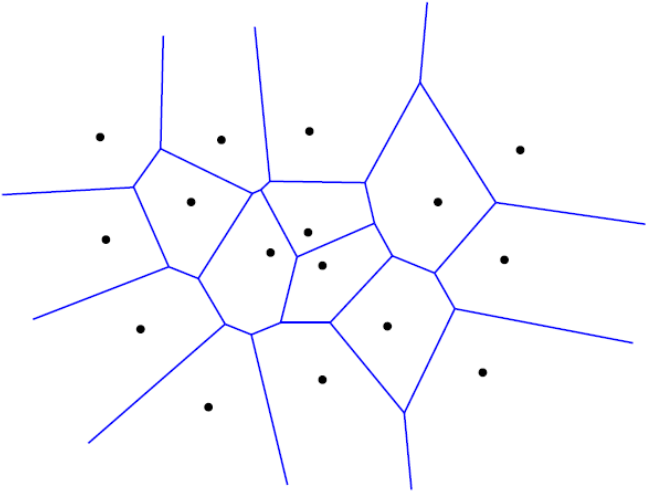
What is Computational Geometry?

- The systematic study of algorithms and data structures to solve geometric problems
- Focus on exact algorithms that are asymptotically fast
  - Data sets can be incredibly large
  - Input sizes grow exponentially as the number of dimensions increases
- Gain new insights from reformulating a problem in geometric terms
  - Querying a database: determine set of points contained in n-dimensional cube
- Many other application areas: robotics, computer graphics...

A Geometric Problem

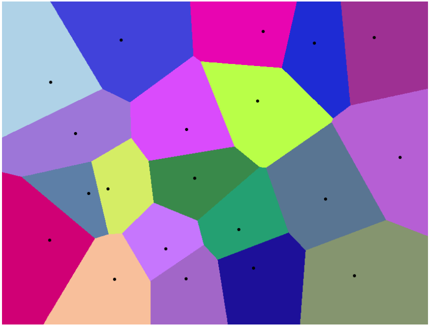


A Geometric Problem



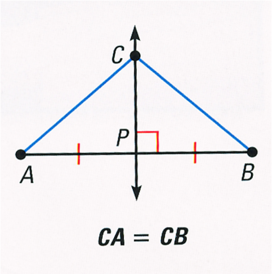
Definitions

- Sites: Distinct central places or points of interest
  - Denote the set of n sites by $P := \{p_1, p_2, \ldots, p_n\}$
- Voronoi diagram: The subdivision of the plane into n cells such that a point q lies in the cell corresponding to p_i iff $dist(q, p_i) < dist(q, p_j)$ for each $p_j \in P$ with $j \neq i$



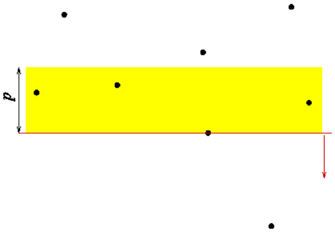
A Naïve Algorithm

- The set of all points equidistant from two sites p_i and p_j is the perpendicular bisector of the line segment connecting p_i and p_j
- Each Voronoi cell is the intersection of $n-1$ half-planes induced by the perpendicular bisectors between p_i and all other sites in P
- Unfortunately, computing the diagram this way runs in $O(n^2 \log n)$ time...



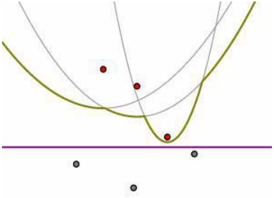
Plane Sweep Paradigm

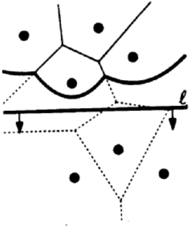
- Imagine sweeping a horizontal line from top to bottom over the plane
- Everything above the sweep line has been computed, while everything below has not
- Maintain information about the intersection of our structure with the sweep line
  - Event points: Locations where this information changes



Beach Line

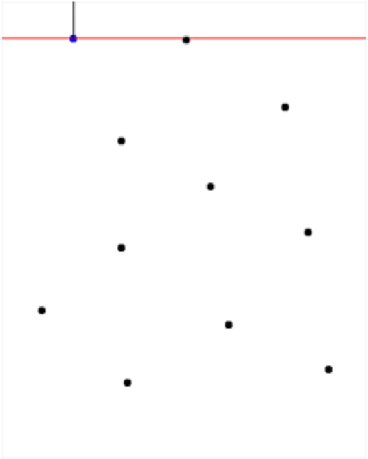
- The Voronoi diagram above the sweep line is affected by event points below the line
- We know the nearest site of a point q if q lies at least as close to a site as it does to the sweep line
- The set of points equidistant from a point and a line defines a parabola
- Beach line: The sequence of parabolic arcs closest to the sweep line





Break Points

The break points of the beach line trace out the edges of the Voronoi diagram as the sweep line moves



Source: *Wikipedia*

Event Points

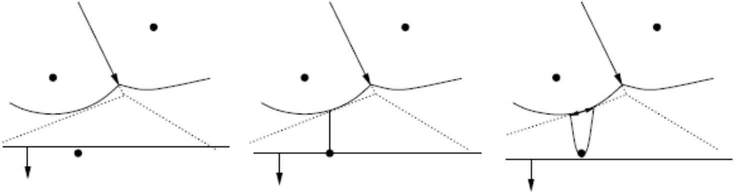
- Site event: Correspond to adding a new edge to the diagram
  - These are our predetermined events
- Circle event: Correspond to adding a new vertex to the diagram
  - Dynamically added as we process site events

Fortune's Algorithm

Algorithm VORONOIDIAGRAM(P)
Input. A set P of point sites in the plane.
Output. The Voronoi diagram given inside a bounding box.

1. Initialize an event queue Q with all site events.
2. **while** Q is not empty
3. **do** Remove the event with the largest y-coordinate from Q.
4. **if** the event is a site event
5. **then** HANDLESITEEVENT(p_i)
6. **else** HANDLECIRCLEEVENT(γ)
7. Compute a bounding box that contains all vertices of the Voronoi diagram in its interior

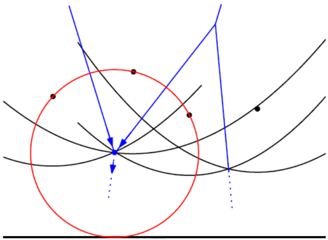
Handing Site Events



- A new parabola is added to the beach line
- New break points begin to trace out the same new edge
- Check the new triple of consecutive arcs for any potential circle events

Handling Circle Events

- Store a circle event as the lowest point on an empty circle containing 3 or more sites
- When the sweep line reaches a circle event, a parabola disappears from the beach line
- The center of this circle is added as a vertex of the Voronoi diagram



Conclusion

Theorem

The Voronoi diagram of n point sites can be computed in $O(n \log n)$ time using Fortune's algorithm

References

de Berg, Cheong, van Kreveld, Overmars
Computational Geometry: Algorithms and Applications
Springer, 2008.

David M. Mount
CMSC754 Computational Geometry Lecture Notes
p. 67 - 74