Voronoi Diagrams

Swami Sarvottamananda

Ramakrishna Mission Vivekananda University

BHU-IGGA, 2010

Outline I

Applications in Cross-disciplines

This concept has independently emerged, and proven useful, in various fields of science.
Different names particular to the respective field have been used, such as *medial axis transform* in biology and physiology, *Wigner-Seitz zones* in chemistry and physics, *domains of action* in crystallography, and *Thiessen polygons* in meteorology and geography.

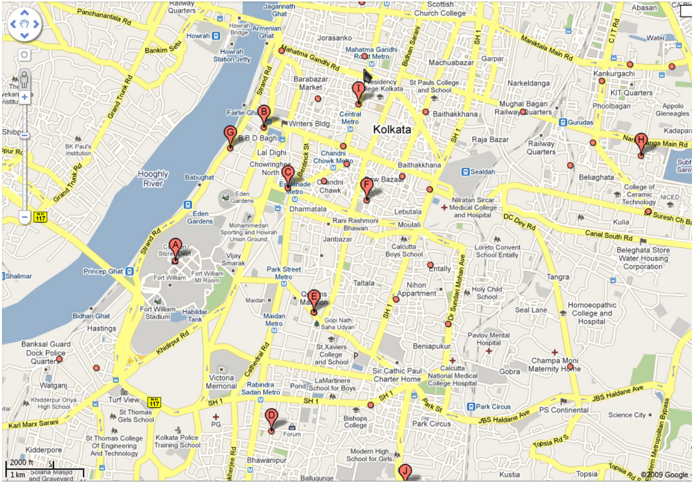
Motivation

Problem (Voronoi Diagrams — The Post Office Problem)

This problem is mentioned in Chapter 7 of the book by de Berg et al. [3]

In a city with several post offices we would like to mark the service region of each post office by proximity.

Post-offices in a section of Kolkata

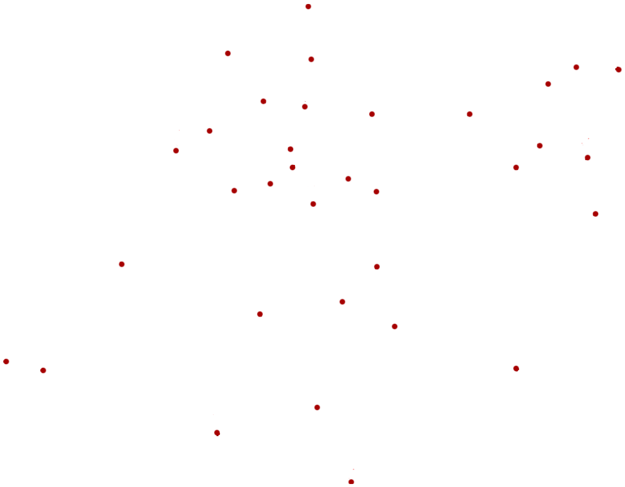


Postoffice Problem and its Simplification

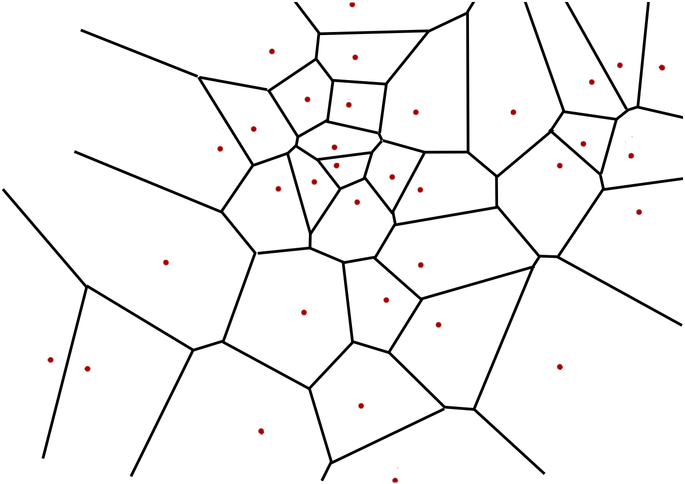
We would like to demarcate the service regions of the post-offices.

To simplify, let us only consider the Euclidean distance and not the actual distance by the roads.

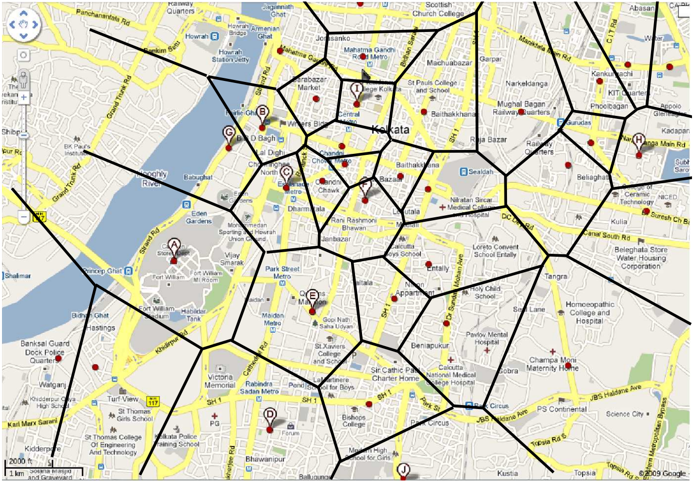
Post-offices as Points in Plane



Proximity Regions of Postoffices

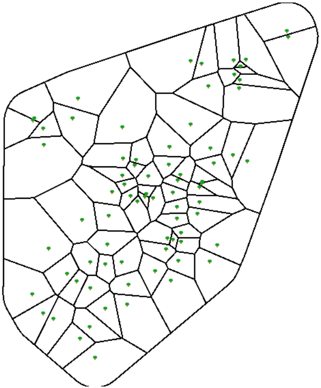


Postoffice Services in Kolkata



Other Direct Applications—I

Example (Jurisdiction of Schools in Boston)

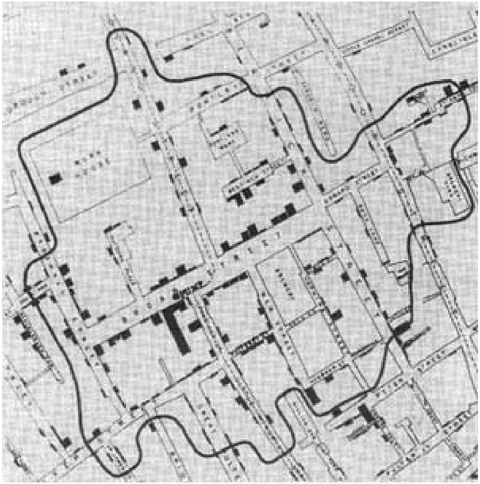


Other Direct Applications—II

Example (1854 Cholera Epidemic in London)

A particularly notable use of a Voronoi diagram was the analysis of the 1854 cholera epidemic in London, in which physician John Snow determined a strong correlation of deaths with proximity to a particular (and infected) water pump on Broad Street.

John Snow's Voronoi Diagram



Voronoi Diagrams

Voronoi Diagrams in Internet

Where you can play around with Voronoi Diagrams—

- Java application at *http://www.nirarebakun.com/voro/ehivorocli.html* (includes higher order VD's),
- VoroGlide (another Java application) *http://wwwpi6.fernuni-hagen.de/GeomLab/VoroGlide/*
- Anything related to Computational Geometry, start at Jeff Erickson's web page, *http://compgeom.cs.uiuc.edu/ jeffe/compgeom/compgeom.html*
- And, much more.

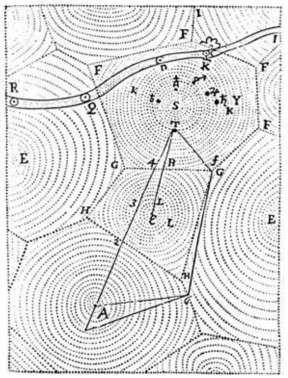
(All the hyperlinks alive on January 2010)

Origin of Problem

- In *Principles of Philosophy* [4] by R. Descartes, 17th century.
- Decomposition of space into convex regions of influence of heavenly bodies.

Origin of Problem

Example (Descartes Explanation of Gravity)



What are Voronoi Diagrams

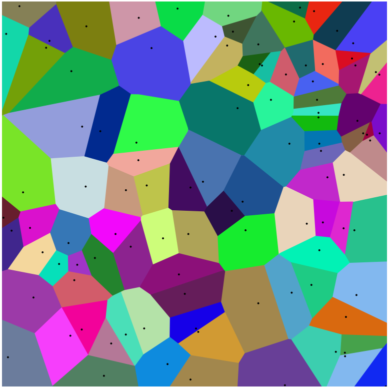
- What are *Voronoi Diagrams*?

What are Voronoi Diagrams

- What are *Voronoi Diagrams*?
- Informally, Voronoi Diagrams are proximity regions to a set of objects.

Voronoid Diagram

Example (Voronoi Diagram in Plane)



Proximity Query

- A dataset S of n points called *sites* in $\Re^2$.
  Let $S = \{s_1, s_2, \ldots, s_n\}$.

Proximity Query

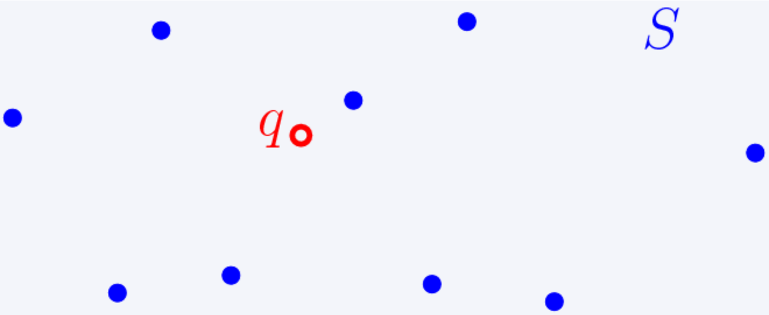
- A dataset S of n points called *sites* in $\Re^2$.
  Let $S = \{s_1, s_2, \ldots, s_n\}$.
- For any given *query point* find the closest site to q.

Proximity Query

- A dataset S of n points called *sites* in $\Re^2$.
  Let $S = \{s_1, s_2, \ldots, s_n\}$.
- For any given *query point* find the closest site to q.
- Find the polygonal *proximity regions* of all sites.

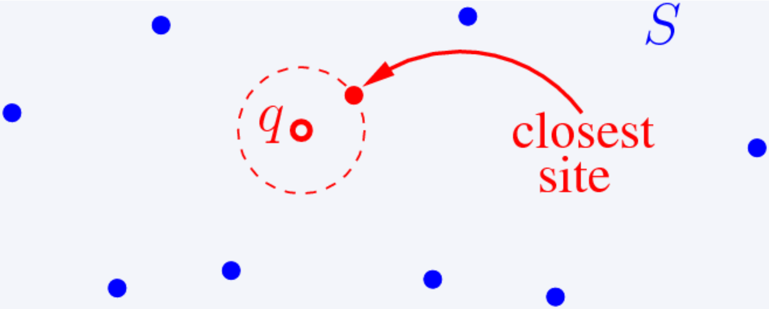
Proximity Query — Illustration

Example (Sites and Query Point)



Proximity Query — Illustration

Example (Nearest Site to Query Point)



Proximity Query — Easy Solution

- Naturally the solution can be found in linear time.
  Just calculate the distances from all sites and choose the closest site.

Proximity Query — Easy Solution

- Naturally the solution can be found in linear time.
  Just calculate the distances from all sites and choose the closest site.
- However, if you have to do it repeatedly, say m times, you can only do it in $O(nm)$-time—quadratic time.

Proximity Query — Easy Solution

- Naturally the solution can be found in linear time.
  Just calculate the distances from all sites and choose the closest site.
- However, if you have to do it repeatedly, say m times, you can only do it in $O(nm)$-time—quadratic time.
- Or, if you have to find the *proximity regions* themselves, the easy algorithm does not work.

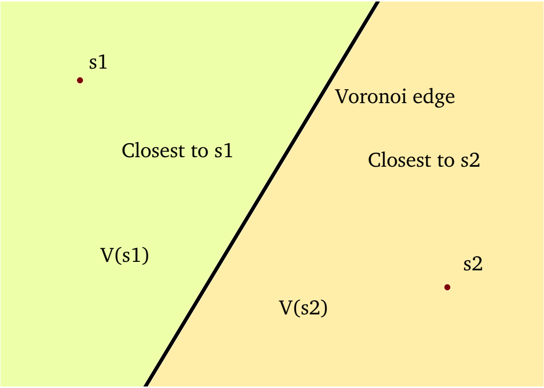
What are Voronoi Diagrams

Definition

- Let S be a set of n distinct points, $foralli \in n, s_i$, (sites) in the plane
- The Voronoi diagram of S is the subdivision of the plane into n cells, one for each site
- A point q lies in the cell corresponding to a site $s_i \in S$ iff $||q - s_i|| < ||q - s_j||$, for each $s_j \in S, i \neq j$
- Cell of s_i is denoted by $V(s_i)$

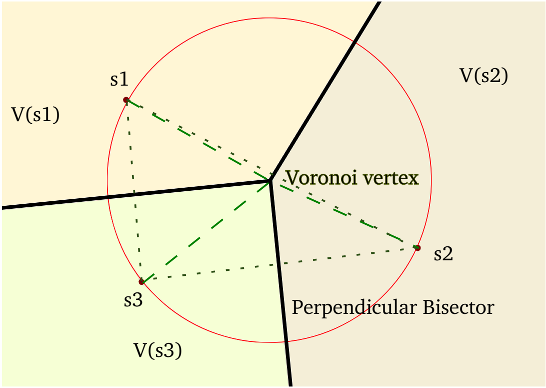
Computing Voronoi Diagram

Example (Voronoi Diagram of 2 points)



Properties of Voronoi Diagram

Example (Voronoi Diagram of 3 points)



Properties

- Voronoi edges are in part perpendicular bisectors.

Properties

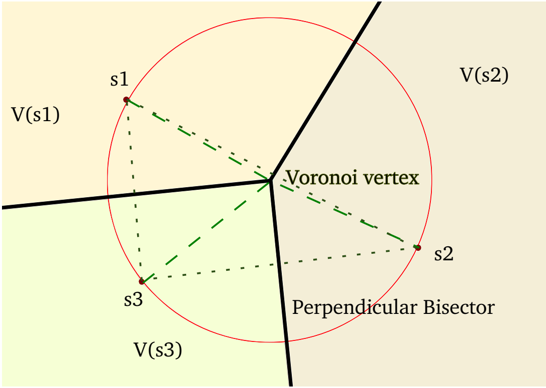
- Voronoi edges are in part perpendicular bisectors.
- Voronoi vertices are equidistant to three points (circumcentre).

Properties

- Voronoi edges are in part perpendicular bisectors.
- Voronoi vertices are equidistant to three points (circumcentre).
- Not obvious, the circumcircle is empty.

Properties of Voronoi Diagram

Example (Voronoi vertex is circumcentre)



Circumcircle is empty and voronoi vertex in centre of circumcircle

Straight-forward Algorithm

Frontal Attack to the Problem

- Compute ${}^nC_3$ circumcentres
- Remove non-empty circumcircles
- Intelligently join circumcentres by bisectors
- Voila, you get the Voronoi Diagram

Bad algorithm, $O(n^4)$ time complexity

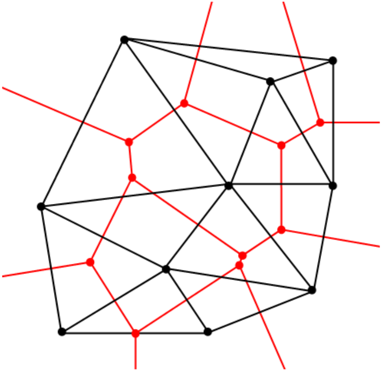
Space Complexity of Voronoi Diagrams

Before giving good algorithms, what is the size of output.

- Number of Voronoi Vertices is $O(n)$
- Number of Voronoi Edges is $O(n)$
- So, Size of Voronoi Diagrams is $O(n)$

Proof of Space Complexity

We look at Dual of Voronoi Diagram (Delaunay Triangulation)



Voronoi vertices are Delaunay faces, Voronoi edges are Delaunay edges

Proof of Space Complexity

- Euler's relation for planar graph is $v + f = e + 2$

Proof of Space Complexity

- Euler's relation for planar graph is $v + f = e + 2$
- $e \leq 3v - 6$, again from Euler, and therefore,

Proof of Space Complexity

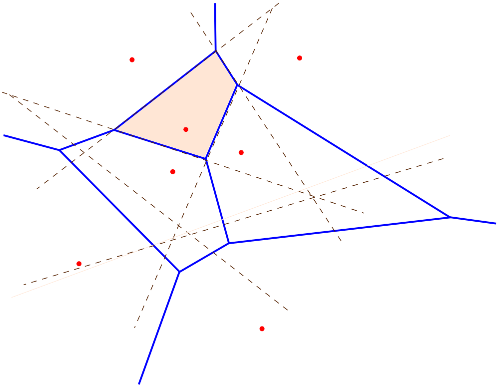
- Euler's relation for planar graph is $v + f = e + 2$
- $e \leq 3v - 6$, again from Euler, and therefore,
- $f \leq 2v - 4$

Proof of Space Complexity

- Euler's relation for planar graph is $v + f = e + 2$
- $e \leq 3v - 6$, again from Euler, and therefore,
- $f \leq 2v - 4$
- Since there is an implicit face at ∞, which corresponds to no voronoi vertex, number of voronoi vertices $\leq 2v - 5$.

Another Algorithm

Each Voronoi cell is intersection of $n-1$ half planes



Another Algorithm—Analysis

- Each $V(s_i)$ is intersection of $n-1$ half planes

Another Algorithm—Analysis

- Each $V(s_i)$ is intersection of $n-1$ half planes
- Intersection of n halfplanes can be computed in $O(n \log n)$ time

Another Algorithm—Analysis

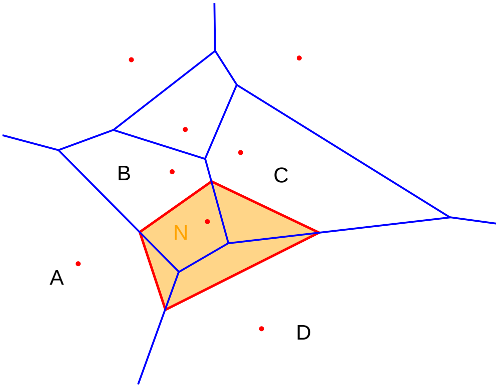
- Each $V(s_i)$ is intersection of $n-1$ half planes
- Intersection of n halfplanes can be computed in $O(n \log n)$ time
- Voronoi diagram of S can be computed in $O(n^2 \log n)$ time by computing the Voronoi cells one by one.

Another Algorithm—Analysis

- Each $V(s_i)$ is intersection of $n - 1$ half planes
- Intersection of n halfplanes can be computed in $O(n \log n)$ time
- Voronoi diagram of S can be computed in $O(n^2 \log n)$ time by computing the Voronoi cells one by one.
- A large improvement over previous method

Yet Another Algorithm

Each point is inserted in Voronoi Diagram one by one



Yet Another Algorithm

- Site s_{i+1} is located in Voronoi Diagram of i sites

Yet Another Algorithm

- Site s_{i+1} is located in Voronoi Diagram of i sites
- Its boundary is calculated

Yet Another Algorithm

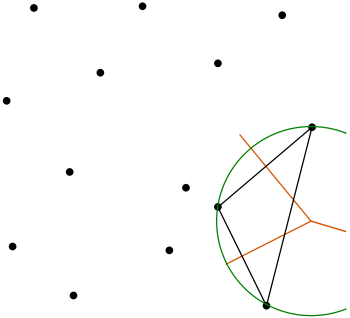
- Site s_{i+1} is located in Voronoi Diagram of i sites
- Its boundary is calculated
- Voronoi diagram of S can be computed in $O(n^2)$ as $O(i)$ edges might be added in each step

Yet Another Algorithm

- Site s_{i+1} is located in Voronoi Diagram of i sites
- Its boundary is calculated
- Voronoi diagram of S can be computed in $O(n^2)$ as $O(i)$ edges might be added in each step
- Further improvement over previous method

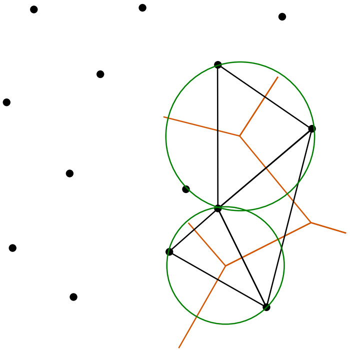
One more Incremental Algorithm-I

Each Voronoi vertex is inserted in Voronoi Diagram one by one



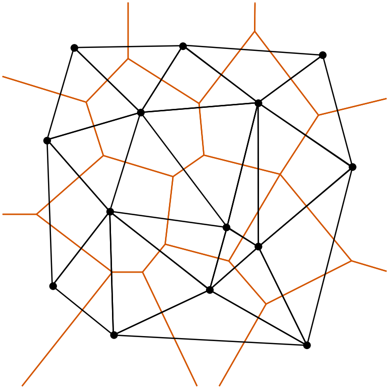
We start with any hull edge and compute *the* empty circle

One more Incremental Algorithm-II



Compute adjacent empty circles

One more Incremental Algorithm-III



Final Voronoi Diagram

Complexity Analysis of Incremental Algorithm

This algorithm also computes Voronoi Diagram in $O(n^2)$

Even though the discussed algorithms are non-aptimal, they are useful for Voronoi Diagram generalisations where the structures are complex and cannot be calculated using usual methods

Can we Compute Voronoi Diagram Faster?

- Clearly no method is optimal.
- Can we do better?

Fortune's Algorithm

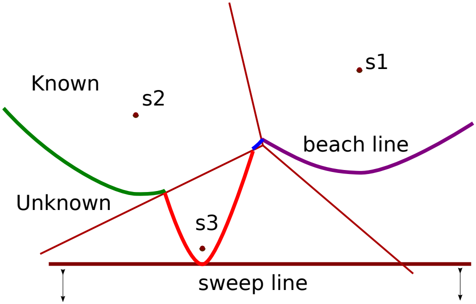
Assumptions

General position assumption: No four sites are co-circular.

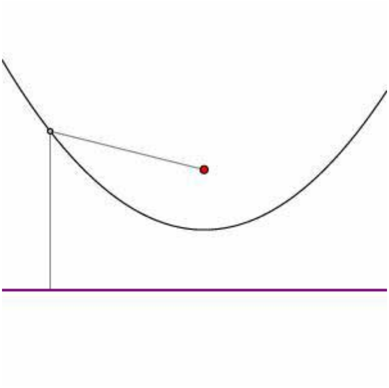
Figures sources:
http://www.ams.org/featurecolumn/archive/voronoi.html

Tentative Idea

Fortune's algorithm is sweep line method [6]

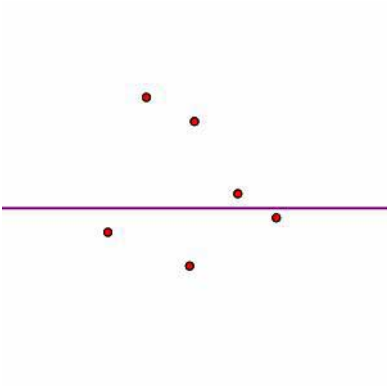


What is Beach line-I



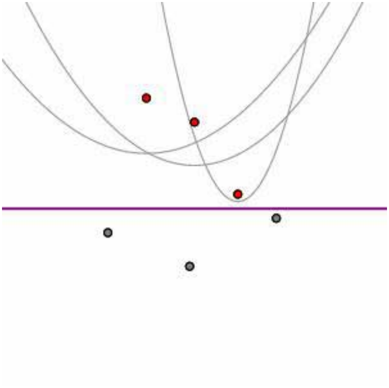
We know equidistant points from a site and a line is a parabola

What is Beach line-II



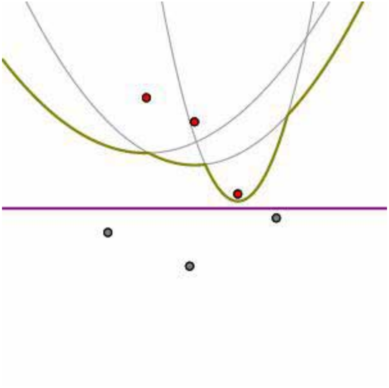
Suppose we have got six sites

What is Beach line-III



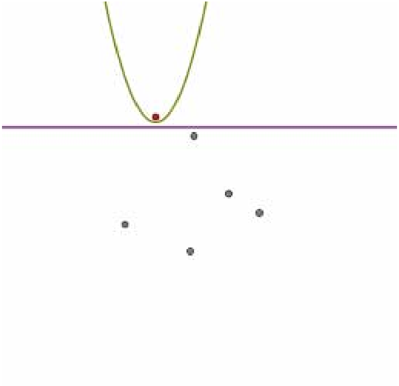
Three sites are already seen by sweep line

What is Beach line-IV

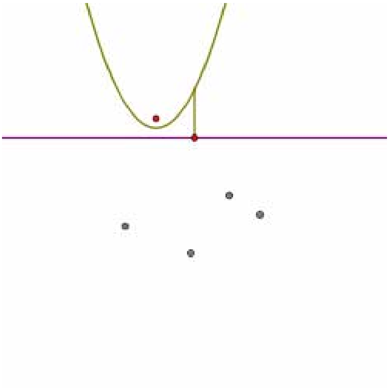


Beach line will be the envelope of the parabolae

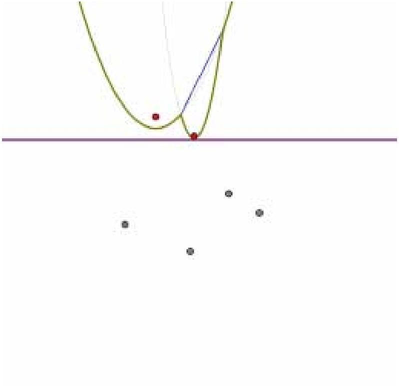
Initial Situation



A site is encountered

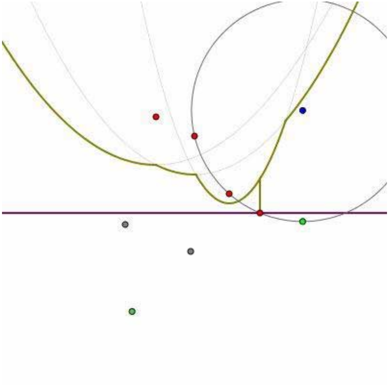


Enter a Voronoi Edge



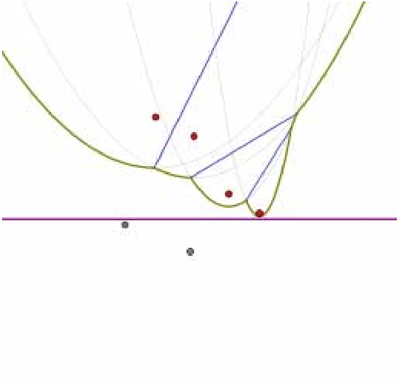
That means we have a Voronoi edge

Enter a Voronoi Vertex



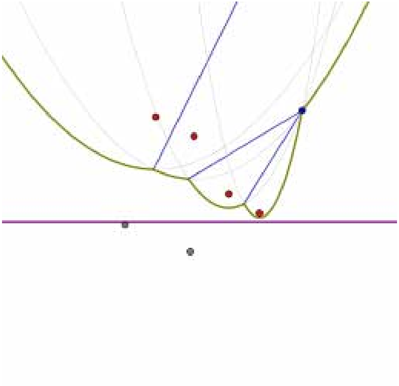
For a Voronoi Vertex we need to add an event (circle event)

Circle Event-I



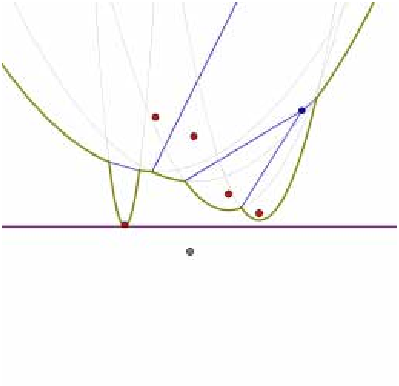
Before circle event

Circle Event-II



At the circle event

Circle Event-III

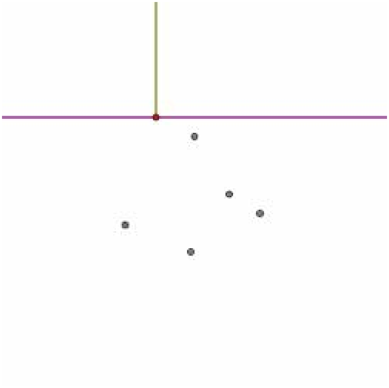


After circle event

Execution of the Fortune's Algorithm

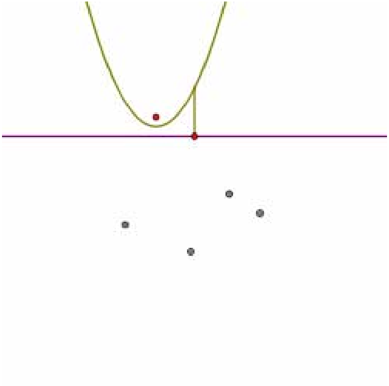
Let us see how the Fortune's algorithm calculates the Voronoi Diagram step by step

Algorithm—Step 1



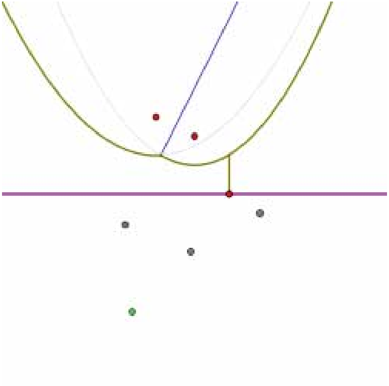
First site and a strange beach line

Algorithm—Step 2



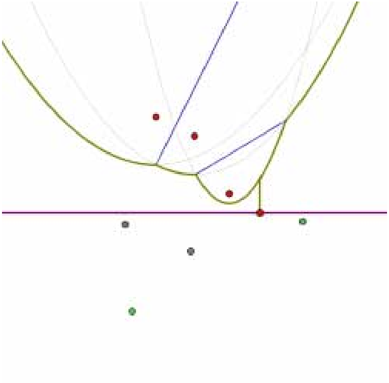
Second site

Algorithm—Step 3



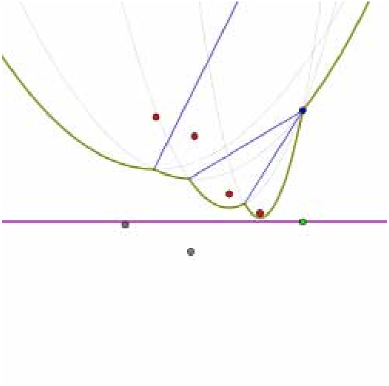
Third site

Algorithm—Step 4



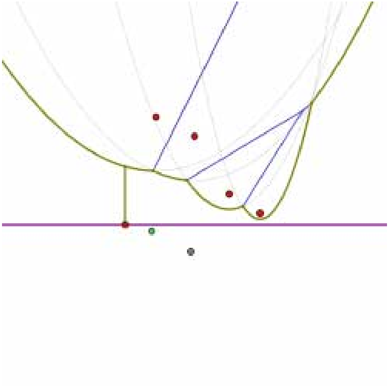
Fourth site and a circle event

Algorithm—Step 5



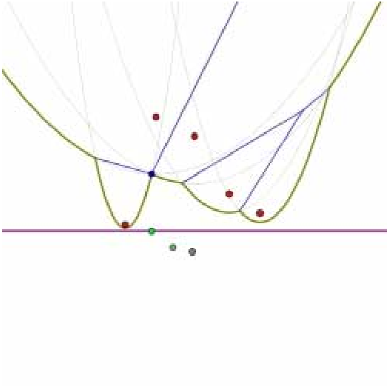
Circle event

Algorithm—Step 6



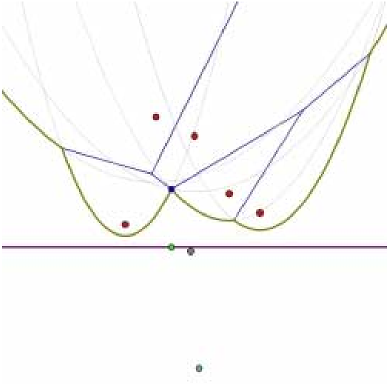
Fifth site and a circle event

Algorithm—Step 7



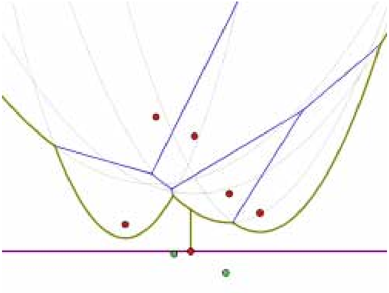
A circle event and another circle event

Algorithm—Step 8



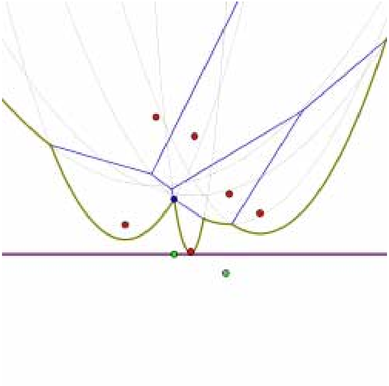
Another of the same

Algorithm—Step 9



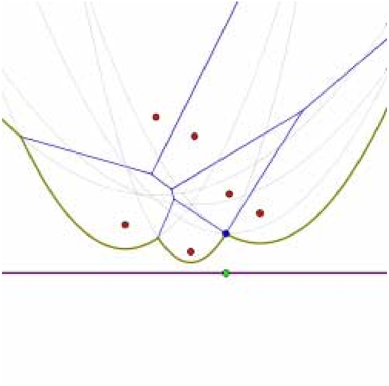
Last site and two circle event

Algorithm—Step 10



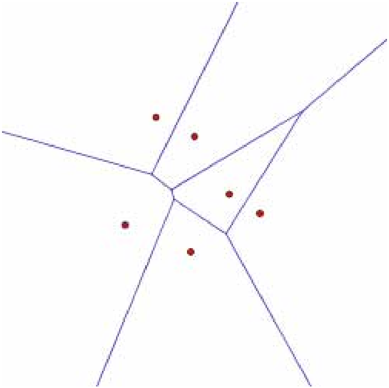
Last but one circle event

Algorithm—Step 11



Last circle event

Algorithm—Step 12



Final Output

Analysis of the Fortune's Algorithm

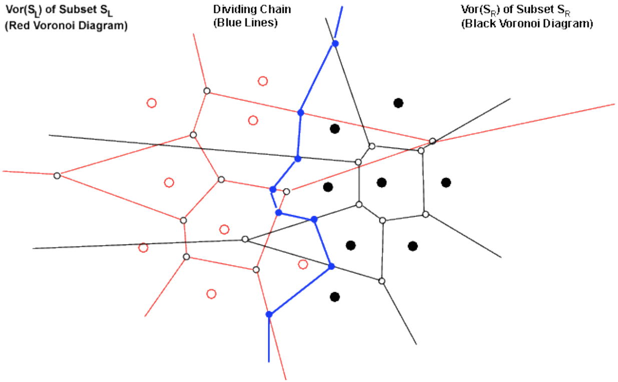
Fortune's algorithm being an example of typical sweep line technique is $O(n \log n)$

Optimal because sorting problem can be reduced to construction of Voronoi Diagrams problem

Divide and Conquer

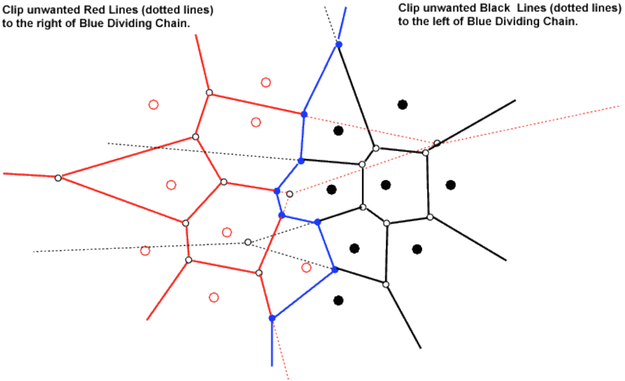
Another algorithm that gives Voronoi Diagram in optimal time and space

Divide



Solve two subproblems

Conquer



Sew the two jumbled pieces

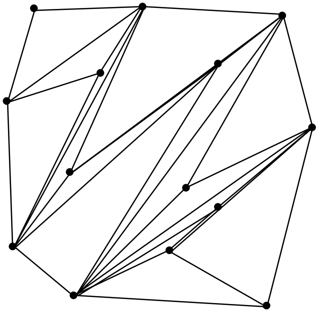
Delaunay Triangulations

What are Delaunay Triangulation?

- Given a set of points sometimes it is needed to join them in triangles

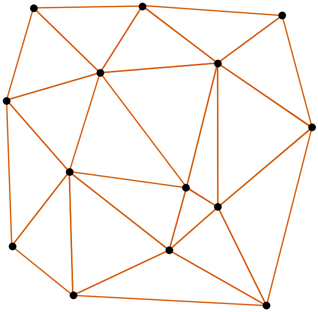
Bad triangulation

Example (Skinny Triangles)



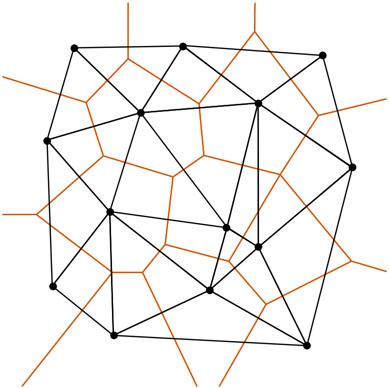
Good triangulation

Example (Delaunay Triangulation)



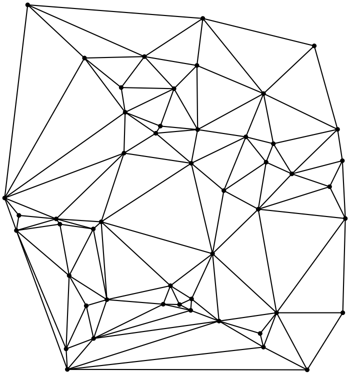
Delaunay triangulation as Dual of Voronoi Diagrams

Example (Delaunay Triangles are Voronoi Vertices)



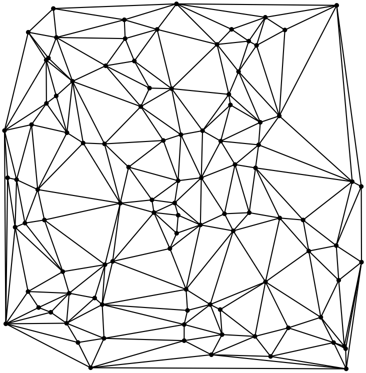
Another Example to Convince

Example (Delaunay Triangulation of 50 points)



Yet Another Example to Convince

Example (Delaunay Triangulation of 100 points)

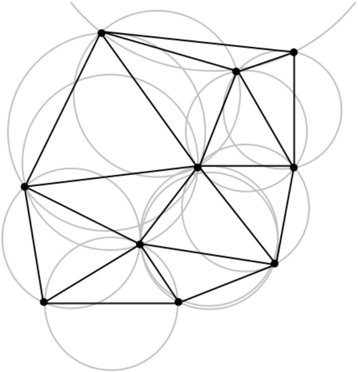


What are we looking for?

- Triangles should not be very stretched (skinny triangles)
- Maximize the minimum angle of all the angles of the triangles in the triangulation
- Circumcircles of all the triangles in the net are empty

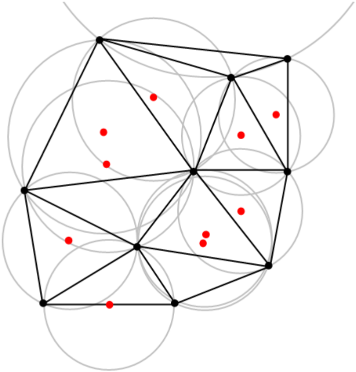
Delaunay Triangulation Properties-I

Example (Delaunay Triangles have Empty Circumcircles)



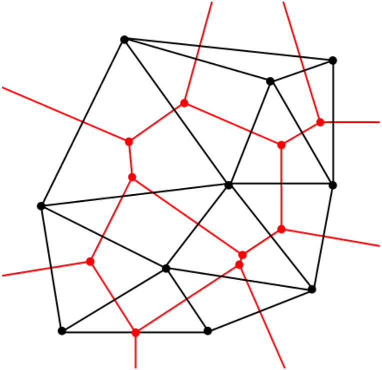
Delaunay Triangulation Properties-II

Example (Empty Circumcircles centred at Voronoi Vertices)



Delaunay Triangulation Properties-II

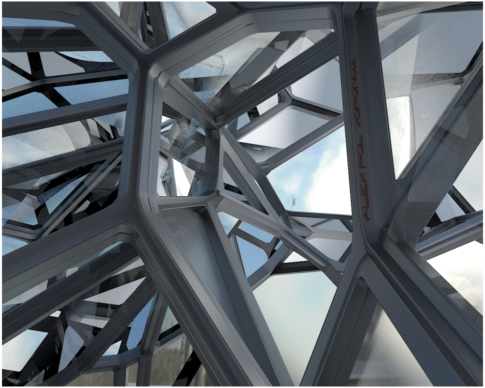
Example (Empty Circumcircles—Dual of Voronoi Diagrams)



Generalisations

Higher Dimension Voronoi Diagram

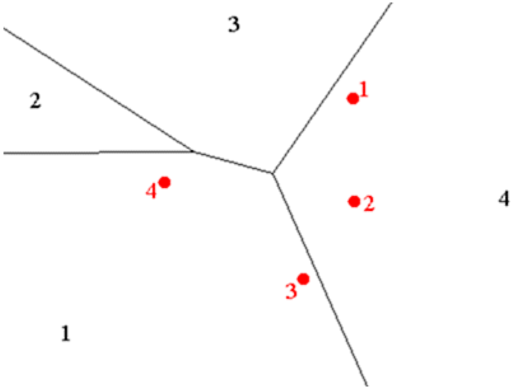
Example (Voronoi Region 3D point set)



Voronoi Diagram in space

Farthest Point Voronoi Diagram

Example (Voronoi regions for farthest point queries)

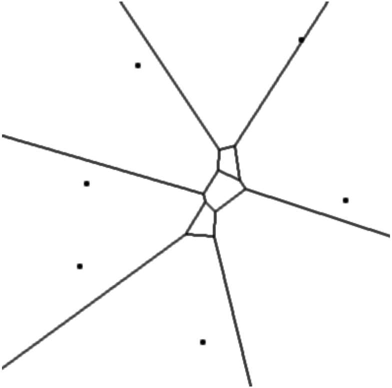


Used for Min-Max optimisation problems

Gives you minimum enclosing circle directly. Though linear time algorithms that do not use farthest point Voronoi Diagram are known.

Higher Order Voronoi Diagram

Example (Voronoi Region for k-nearest point set)



Second order Voronoi Diagram

Higher-order Voronoi diagrams can be generated recursively.

To generate the n^{th}-order Voronoi diagram from set S, start with the $(n1)^{th}$-order diagram and replace each cell generated by $X = \{x_1, x_2, ..., x_{n1}\}$ with a Voronoi diagram generated on the set SX.

Conclusion

Problems to Ponder

- Compute Voronoi diagram of a set of lines (or line segments) in three dimensions, conjectured to be near quadratic. (Polyhedra is hard)
- What is the maximum number of combinatorial changes possible in a Euclidean Voronoi diagram of a set of n points each moving along a line at unit speed in two dimensions?
- Where to place a site to maximize its Voronoi cell?

Summary

- We studied the concept of Voronoi Diagrams
- Next we saw how they can be computed
- We looked into their duals—Delaunay Triangulations
- Finally we saw some of the generalisations

You may read

- Book by Okabe et al. [7], who lists more than 600 papers, and
- the surveys by Aurenhammer [1],
- Bernal [2], and
- Fortune [6] for a complete overview,
- Also, Chapters 5 and 6 of Preparata and Shamos [8], and
- Chapter 13 of Edelsbrunner [5]

References I

Franz Aurenhammer.
Voronoi diagrams - a survey of a fundamental geometric data structure.
ACM Comput. Surv., 23(3):345–405, 1991.

Javier Bernal.
Bibliographic notes on voronoi diagrams.
Technical report, Natl. Inst. of Standards and Tech., April 1993.
NIST Internal Report 5164.

Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf.
Computational Geometry: Algorithms and Applications.
Springer-Verlag, second edition, 2000.

References II

Rene Descartes.
Principia Philosophiae.
Ludovicus Elzevirius, Amsterdam, 1644.

Herbert Edelsbrunner.
Algorithms in Combinatorial Geometry.
Springer-Verlag, New York, 1987.

Steven Fortune.
Voronoi diagrams and Delaunay triangulations.
CRC Press, Inc., Boca Raton, FL, USA, 1997.

References III

- Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu.
  Spatial tessellations: Concepts and applications of Voronoi diagrams.
  Probability and Statistics. Wiley, NYC, 2nd edition, 2000. 671 pages.

- F. P. Preparata and M. I. Shamos.
  Computational Geometry: An Introduction.
  Springer-Verlag, 1985.

At Last . . .

Thank You

shreesh@rkmvu.ac.in
sarvottamananda@gmail.com