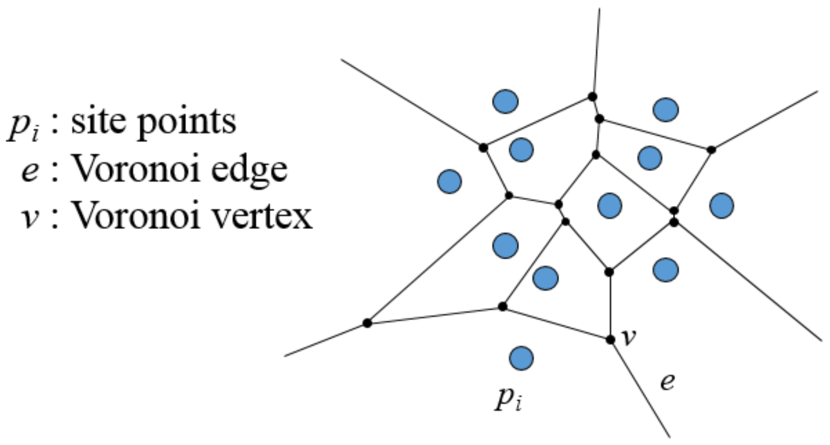
Tessellation Algorithms: Voronoi Diagrams & Delaunay Triangulation

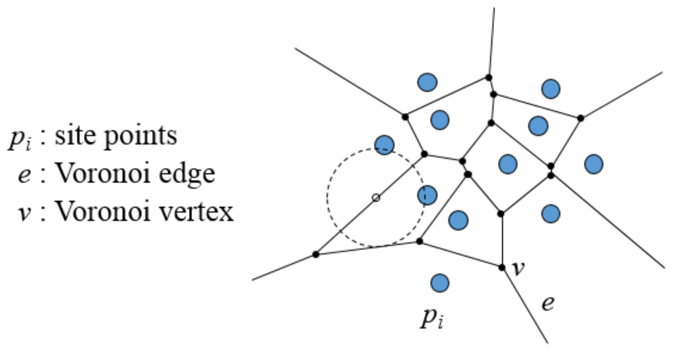
Voronoi Diagram

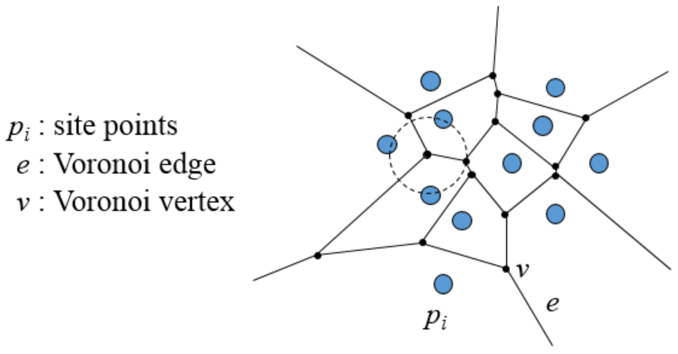
The partitioning of a plane with points into convex polygons such that each polygon contains exactly one generating point and every point in a given polygon is closer to its generating point than to any other.



Voronoi edge :- A point q lies on a Voronoi edge between sites pi and pj iff the largest empty circle centered at q touches only pi and pj. A Voronoi edge is a subset of locus of points equidistant from pi and pj

Voronoi vertex :- A point q is a vertex iff the largest empty circle centered at q touches at least 3 sites. A Voronoi vertex is an intersection of 3 more segments, each equidistant from a pair of sites



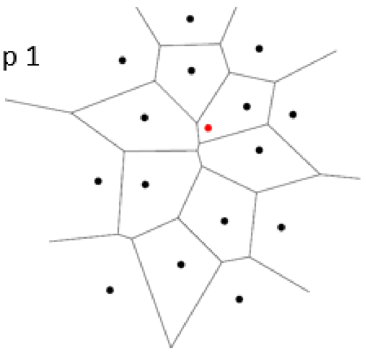


Incremental Algorithm

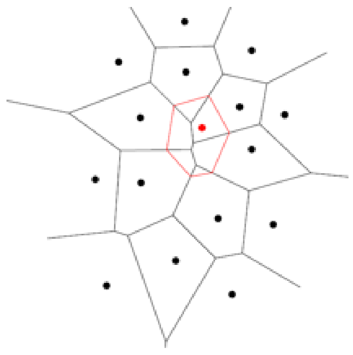
Each point is inserted in Voronoi Diagram one by one.

1. Starting with the Voronoi diagram of {p1…..pi} add point pi+1
2. Explore all candidates to find the site pj closest to pi+1 and compute its region by building its boundary starting from bisector bi+1,j .
3. Update the diagram by pruning the initial edges.

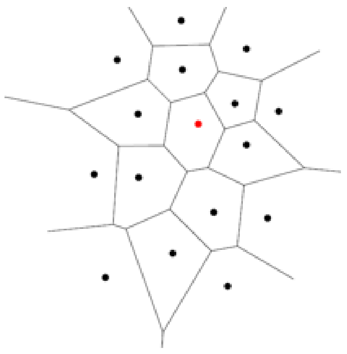
Step 1



Step 2



Step 3

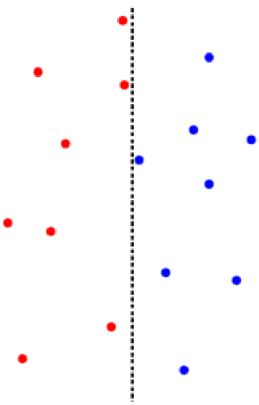


Step 1- Adding a Point
Step 2- Computing its region
Step 3- Prune the initial diagram

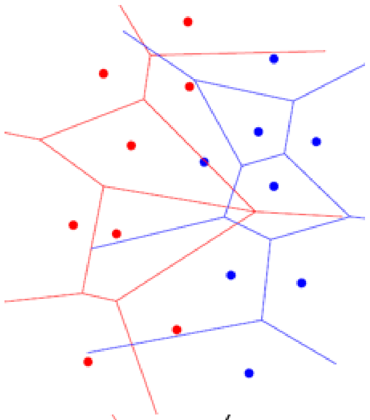
Divide and Conquer Algorithm

1. Sort the points of P by abscissa (only once) and vertically partition P into two subsets R and B, of approximately the same size.
2. Recursively compute Vor(R) and Vor(B).
3. Compute the separating chain.
4. Prune the portion of V or(R) lying to the right of the chain and the portion of V or(B) lying to its left.

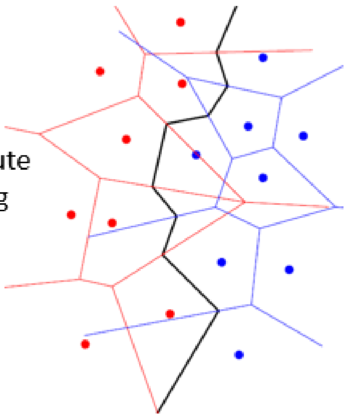
Step1- Divide points into 2 subsets.



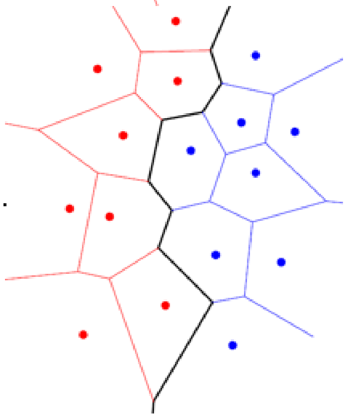
Step2- Draw Voroni Diagram for each set.



Step3- Compute the separating chain.



Step 4- Prune the extra edges.



Fortune's Algorithm (Sweep line)

Voronoi diagram constructed as horizontal line sweeps the set of sites from top to bottom

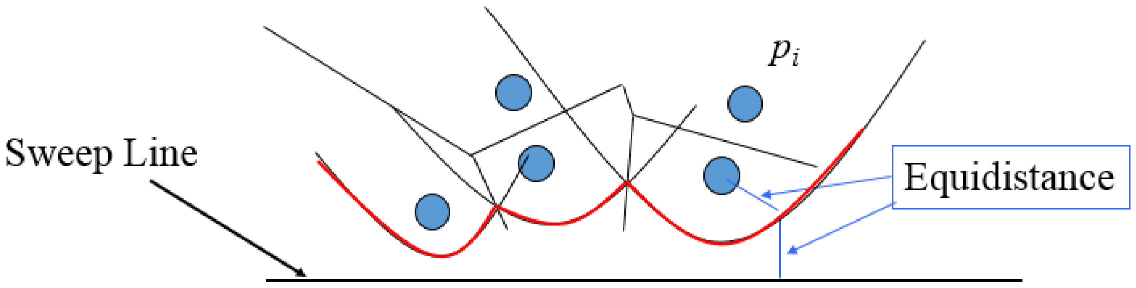
Incremental construction i.e. it maintains portion of diagram which cannot change due to sites below sweep line, keeping track of incremental changes for each site (and Voronoi vertex) it "sweeps"

It maintain a representation of the locus of points q that are closer to some site pi above the sweep line than to the line itself (and thus to any site below the line).

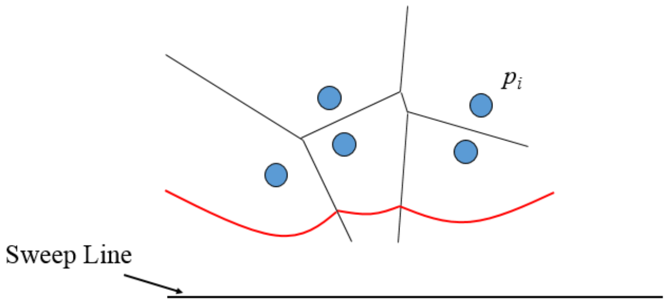
The set of parabolic arcs form a beach-line that bounds the locus of all such points

Sweep line Properties:-

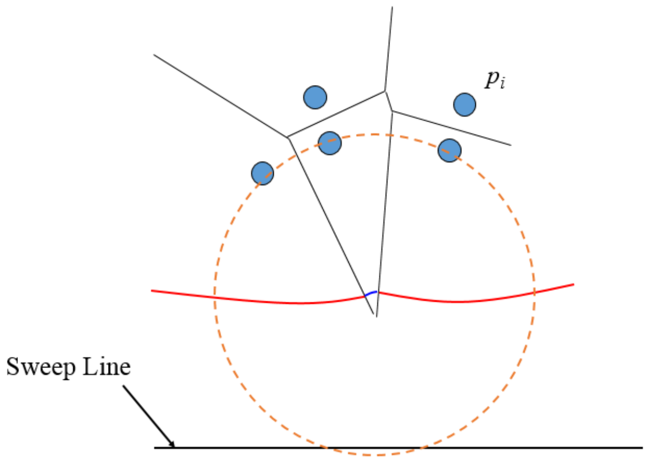
- Voronoi edges are traced by the break points as the sweep line moves down.
- Emergence of a new break point(s) (from formation of a new arc or a fusion of two existing break points) identifies a new edge
- Voronoi vertices are identified when two break points meet (fuse).
- Decimation of an old arc identifies new vertex



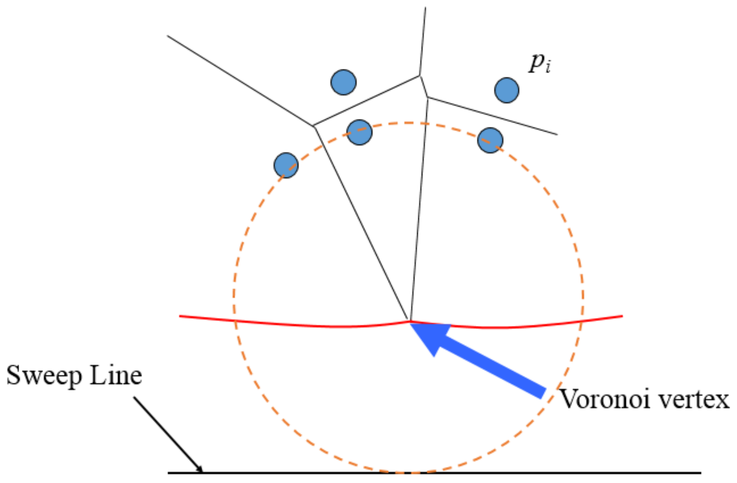
Break points trace out Voronoi edges



Arcs flatten out as sweep line moves down



Eventually, the middle arc disappears



A circle is detected that is empty (contains no sites) and touches 3 or more sites

Asymptotic Complexity

- Incremental Algorithm computes Voronoi Diagram in O (n2).
- Both Divide and Conquer and Fortune's Algorithm displays an asymptotic time complexity of O (n (log n)).

Delaunay Triangulation

Delaunay triangulation is a straight-line dual graph of the Voronoi Diagram of a point set. It maximizes the minimum angle of all the angles of the triangles in the triangulation. By application of Thale's Theorem, we conclude that circumcircles of all the triangles are empty.

Flip Algorithms

If we have any triangulation of the points available, then illegal edges are found using Thale's Theorem and then they are flipped repeatedly until no triangle is non-Delaunay.

Incremental Algorithm

This consists of adding points to the triangulation one by one. It basically consists of two steps-

1. Locate i.e. to find the triangle containing the new point
2. Update i.e. to flip illegal edges till all satisfies the Delaunay criterion.

Divide and Conquer Algorithm

In this algorithm, one recursively draws a line to split the vertices into two sets. The Delaunay triangulation is computed for each set, and then the two sets are merged along the splitting line. The merging which is quite straight forward for 2D scenarios is not so simple in 3D cases. For n dimensional space "DeWall: A fast divide and conquer Delaunay triangulation algorithm" is used

Gift Wrapping Algorithm

This algorithms constructs the Delaunay triangulation by starting with a single Delaunay triangle and then incrementally discovering valid Delaunay triangles, one at a time. Each new triangle is grown from an edge of a previously discovered triangle by finding the site that joins with the endpoints of that edge to form a new triangle whose circumcircle is empty of sites.