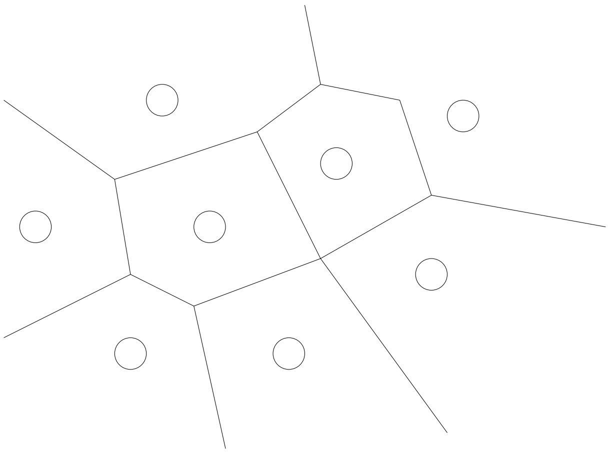
CS 497: Geometrix Data Structures	**Spring 2002**
Lecture 5 — January 29, 2002	
Lecturer: Jeff Erickson	*Scribe: Gregory K. Taylor*

Voronoi Diagram

1. In circle test
2. Depth First Search - $O(n^2)$
3. Divide and Conquer [Shamos* and Hoey]
4. Delannay triangulations
5. Lifting / Duality
6. sweep

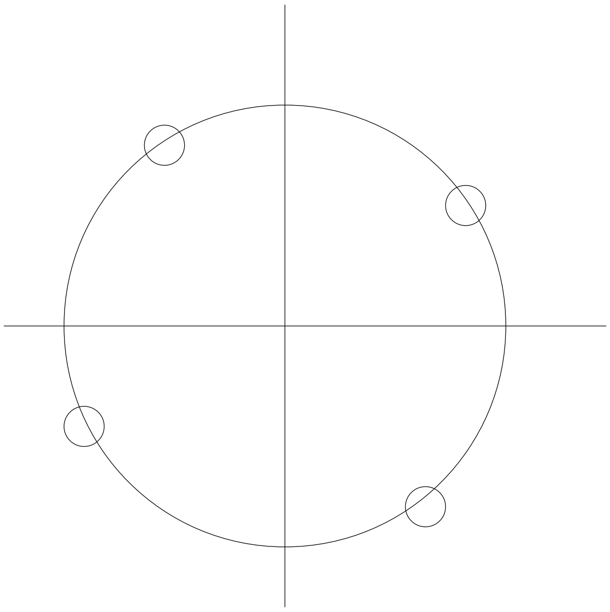
5.1 Voronoi Diagram



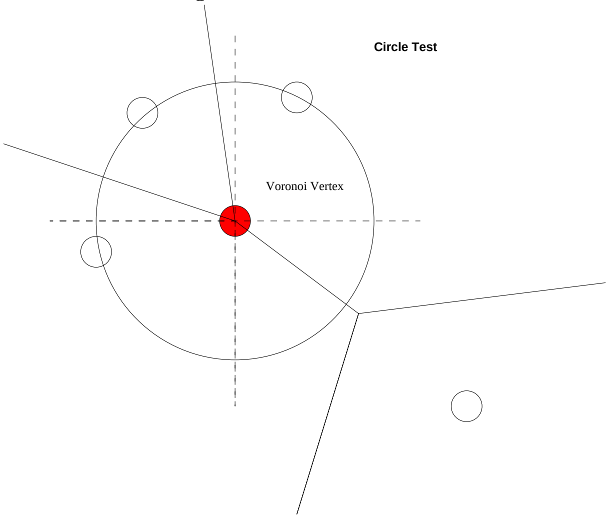
A voronoi diagram may be computed by: Compute one region at a time: $O(n^2 log(n))$ Using Jarvis March: $O(n^2)$

5.2 The in Circle Test

First assume that four points do not lye on the same circle.



We are looking for Vornoi vertecies.

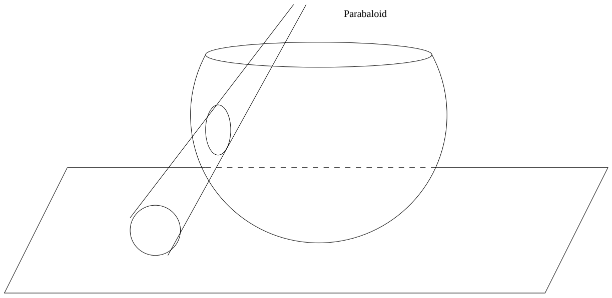


$$\left\|\begin{array}{cccc} 1 & a & b & a^2+b^2 \\ 1 & c & d & c^2+d^2 \\ 1 & e & f & e^2+f^2 \\ 1 & g & h & g^2+h^2 \end{array}\right\| > 0$$

The determinant of this function is greater than 0.

If we evaluate this matrix so that 1, 3, and 2 are in counterclockwise order, then we will be able to find weather the 4th point is outside of the circle or not.

Another way to look at the in circle test is to image a hyperbaloid lying on a plane where a projective transformation is being done on a circle between the two. The circle maps to a plane that disects the hyperbaloid. The in circle test determines which piece the 4th point projects to.



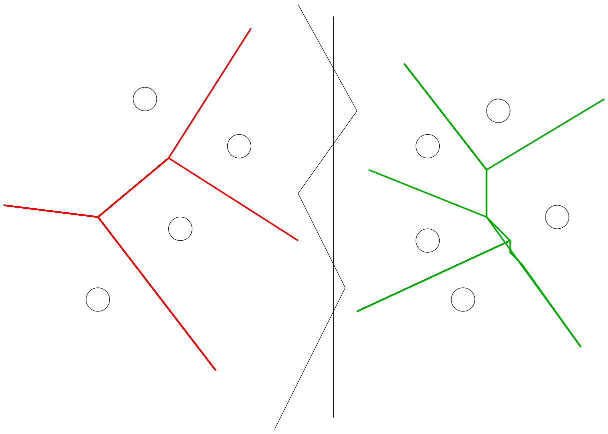
5.3 DFS Vernoi

First, scan left to right to find a vernoi vertex using in circle tests.

```
Find a voronoi edge
  Repeat
    for any unfinished edge
      find its end point (a voronoi vertex)
      using in circle tests
    two new edges.
  until all edges are finished
```

This algorithms runs in $O(n^2)$ time. There are $O(n)$ iterations of the repeat-until loop, and finding an edge's end point requires $O(n)$ time.

5.4 Divide and Conquer



This algorithm roughly divides the plane where the points lie in two. It then computes the vernoi diagram of each half recursivley, and merges the two vernoi diagrams. The left vernoi diagram is red, the right is green.

Shamos Hoey

```
Compute Convex Hull  O(nlogn)
Split into L, R      O(n)
```

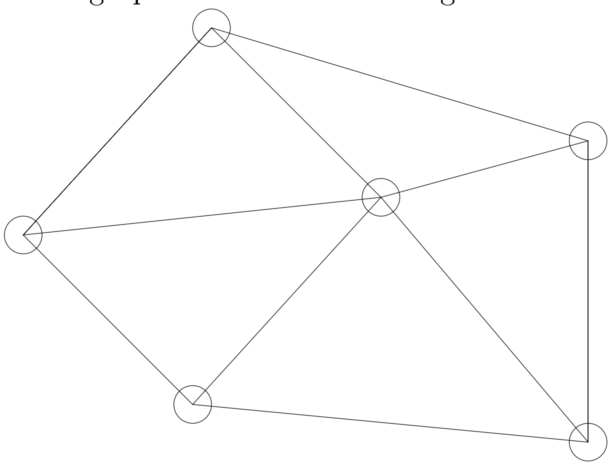
```
Compute V(L), V(R)   O(2T(n/2))
(l,r) = lower common
      tangent of L and R.

repeat    O(n()
  walk along bisector of (l,r)
  until
  hit red edge - change r
  hit green edge, change l
until reach + Infinity
```

The running time of this algorithm is bounded by finding the convex hull in $O(nlog(n))$ time.

5.5 Delannay Triangulation

= planar graph dual of Voronoi diagram



Any 2 3 points form a Delaney edge triangle if they lie on an empty circle.