CISC 868 Fall 2011
Week 6

October 24, 2011

The Voronoi Diagram

We saw this amazing structure and a multitude of different ways in which in can be used to solve problems in computer graphics, urban planning, and mining to name only a few. The Voronoi diagram is the dual graph of the Delaunay triangulation. This means that any algorithm to compute the Delaunay triangulation can be used to obtain the Voronoi diagram with $\mathrm{O}(n)$ overhead. We briefly discussed the algorithm presented in the text to compute the Voronoi diagram directly. It is an $\mathrm{O}(n^2 \log n)$ algorithm and can be summarized as follows.

> Input: A set of planar point sites S.
> Output: The Voronoi diagram of S.
>
> For every site $s \in S$ compute $\mathrm{V}(s)$, the Voronoi cell of s by intersecting $n-1$ half-planes.

The $\mathrm{O}(n^2 \log n)$ complexity can be obtained by intersecting the the $n-1$ halfplanes in $\mathrm{O}(n \log n)$ time. We discussed a divide and conquer approach to accomplishing this.

> Input: P a set of n (possibly unbounded) convex polygons.
> Output: The intersection of the polygons.
>
> If $n = 2$ compute the intersection directly.
> Otherwise divide P into two subsets, P_1 and P_2, of cardinality $\lceil n/2 \rceil$ and $\lfloor n/2 \rfloor$, and recursively compute the intersections of the polygons in P_1 and P_2.

We saw that the intersection of two convex n-gons could be determined in linear time. Thus, we can characterize the complexity of this algorithm using a standard divide and conquer recurrence relation, $F(n) = F(\lceil n/2 \rceil) + F(\lfloor n/2 \rfloor) + g(n)$, where $g(n)$ is a linear function of n. This yields the desired $\mathrm{O}(n \log n)$ result.