Assignment 5 - Computational Geometry (0368-4211)

Due: 9.2.09, in Adam Sheffer's mailbox

Problem 1

Let P be a set of n points in the plane, and let P^* be the lifted set in $\mathbb{R}^3$ (each point $p = (a, b) \in P$ is mapped to the point $p^* = (a, b, a^2 + b^2)$ on the paraboloid $z = x^2 + y^2$ in $\mathbb{R}^3$). Let T be some triangulation of the convex hull of P, which is not Delaunay, and let T^* denote the corresponding lifting of T (each triangle abc of T is lifted to the triangle $a^*b^*c^*$).

(a) Show that there must exist four points a, b, c, d of P, such that (i) abc, abd are triangles in T; (ii) the quadrilateral $acbd$ is convex; and (iii) the two lifted triangles $a^*b^*c^*$, $a^*b^*d^*$ form a concave turn in $\mathbb{R}^3$ (d^* lies below the plane of $a^*b^*c^*$, and c^* lies below the plane of $a^*b^*d^*$). (**Hint:** Use the empty disk condition, and 'lift' it to 3-space.)

(b) Transform T into a new triangulation T_1, by replacing the edge ab by the edge cd. (This is called an *edge flip*.) Show that the graph of the lifted T_1^* lies (weakly) below the graph of T^*.

(c) Use (a) and (b) to show that there exists a sequence of edge flips satisfying the conditions in (a) and (b) which starts at T and terminates at the Delaunay triangulation. Give an upper bound on the number of edge flips in such a sequence.

Problem 2

Let S be a set of n points on the unit sphere in three dimensions (they are all at distance 1 from the origin). Show that (the 3-dimensional) Voronoi diagram $Vor(S)$ has linear complexity, describe its geometric structure, and give an $O(n \log n)$ algorithm for constructing it.

Problem 3

Let S be a set of n pairwise disjoint discs $D_1, \ldots, D_n$ of arbitrary radii in the plane. For a point x not in any of the discs, the distance $d(x, D_i)$ is defined to be the length of the tangent from x to D_i. The Voronoi cell $V(D_i)$ consists of all points x which either lie in D_i or are such that $d(x, D_i) \leq d(x, D_j)$ for all j. The partition of the plane into these Voronoi cells is called the Voronoi diagram of S.

(a) What is the shape of the Voronoi edges? of each Voronoi cell? What is the total complexity of the diagram?

(b) Assume that the plane where S lies is the xy-plane in 3-space. Show that there exists a set S^* of n points in three dimensions, such that (i) each point $p^* \in S^*$ projects vertically to the center of a disk $D \in S$, and (ii) The intersection of the xy-plane with the (standard, Euclidean) 3-dimensional Voronoi diagram of S^* is the Voronoi diagram of S, as defined above.

(c) Give an $O(n \log n)$ algorithm for computing the Voronoi diagram of S.

Problem 4

Let S be a set of n points $s_1, \ldots, s_n$ in the plane. With each point s_i associate a positive weight w_i. Given a point x in the plane, define its distance from s_i to be $d(x, s_i) = w_i |xs_i|$, where $|xs_i|$ denotes the Euclidean distance between these points. The Voronoi cell $V(x_i)$ of x_i is defined as

$$\{x \mid d(x, s_i) \leq d(x, s_j) \text{ for all } j\}$$

and the collection of these cells constitutes the Voronoi diagram of S. Give an example of a set S and weights w_i for which the resulting diagram has $\Omega(n^2)$ complexity.

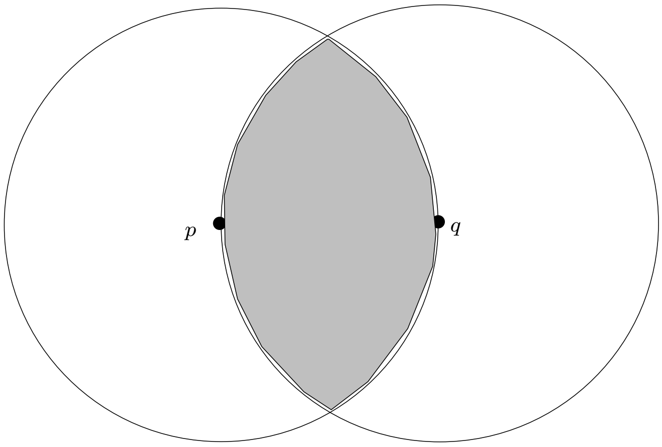
Now let R be the x-axis, and consider the intersection of the Voronoi diagram of S with R. Show that a single Voronoi cell can cross R in many intervals (up to $\Theta(n)$ of them), but the total number of such intervals is at most $2n - 1$. (**Hint**: Show that the Voronoi cell of the site with the largest weight can meet R in at most one interval, and use induction.)

Problem 5

The *Relative Neighborhood Graph* $RNG(S)$ of a set S of points in the plane consists of all edges pq, for $p, q \in S$, such that

$$d(p, q) \leq \min_{r \in S \setminus \{p,q\}} max(d(p, r), d(q, r)).$$

(a) For $p, q \in S$, let $lune(p, q)$ denote the intersection of the two disks centered at p and at q and having radius equal to $d(p, q)$.



Show that $(p, q) \in RNG(S)$ if and only if the interior of $lune(p, q)$ does not contain any point of S.

(b) Show that each edge of $RNG(S)$ is a Delaunay edge.

(c) Design an algorithm to compute $RNG(S)$ efficiently.

(d) Show that the each edge of the Euclidean minimum spanning tree of S is also an edge of $RNG(S)$.