CS 274
Computational Geometry (Spring 2019)
Homework 5

Homework 5 is due **at the start of class (2:10 pm) on Monday, May 6, 2019**.

You may use algorithms learned in class as subroutines without re-explaining them.

[1G] Minkowski sums (6 points). Suppose we want to automatically machine a two-dimensional part by cutting away the region around it with a circular drill bit. Let D be a disk whose center is the origin and whose radius r is the radius of the drill bit. When machining a shape S, we may place the center of the drill bit at any point not in the interior of $S \oplus D$.

We wish to argue that for any n-vertex polygon P (not necessarily convex, possibly with polygonal holes), the worst-case complexity of $P \oplus D$ is in $O(n)$. The complexity is the total number of vertices, line segments, and circular arcs needed to represent the boundary of $P \oplus D$. Line segments are induced by edges of P, offset sideways by the radius r of D. Circular arcs (of radius r) are induced by vertices of P.

As the boundary representation of $P \oplus D$ is a planar graph, it suffices to bound the number of vertices. There are three types of vertices that can appear: intersections between two line segments, intersections between two arcs, and intersections between a line segment and an arc.

(i) Demonstrate that a single edge of P can induce up to $\Theta(n)$ line segments in $P \oplus D$, and that a single vertex of P can induce up to $\Theta(n)$ circular arcs. (A couple of simple figures should suffice to illustrate these two facts—no words are necessary.)

(ii) An edge e of P can induce a sequence of line segments in $P \oplus D$, offset a distance of r from e. Suppose two line segments of $P \oplus D$, s_1 and s_2, intersect at a vertex v. Show that v terminates one of the sequences of line segments induced by some edge of P. Use this fact to show that $P \oplus D$ can have only $O(n)$ vertices where pairs of line segments meet. (For clarity, please use "edge" to denote an edge of P, and "segment" to denote an edge of $P \oplus D$.)

(iii) Show that $P \oplus D$ can have only $O(n)$ vertices where pairs of arcs meet. The easiest way to do this is to explain the relationship between these vertices and the edges of a Voronoi diagram.

Note: the number of segment-arc intersections can be bounded by the same reasoning as part (iii), by using the properties of the Voronoi diagram of the line segments. (See Section 7.3 of the third edition of the Dutch Book if you're curious what a Voronoi diagram of line segments is.)

[2] Subsegment encroachment in Ruppert's algorithm (4 points). Given a vertex set $V \subset \mathbb{R}^2$, show that an edge $e \in \mathrm{DT}(V)$ is encroached if and only if the Delaunay triangulation $\mathrm{DT}(V)$ contains a triangle that has e for an edge and a nonacute angle ($\geq 90°$) opposite e. Therefore, a subsegment's encroachment can be diagnosed in $O(1)$ time.

[3] Strengthening Ruppert's algorithm (4 points). Suppose Ruppert's algorithm enforces a stricter standard of quality for triangles that do not intersect the relative interior of a segment. A triangle that does not intersect a segment interior is split if its circumradius-to-shortest edge ratio exceeds 1 (corresponding to a minimum angle less than 30°). A triangle that intersects a segment interior is split if its circumradius-to-shortest edge ratio exceeds $\sqrt{2}$ (corresponding to a minimum angle less than approximately 20.7°). Given a PSLG in which no two segments meet at an angle less than 90°, show that the algorithm still must terminate.

[4G] Polygonal curve reconstruction (5 points). Let Σ be the boundary of a simple polygon, not necessarily convex, with finitely many boundary edges. Let P be a finite set of points sampled on Σ, and let $\mathrm{DT}(P)$ be its Delaunay triangulation. Note that P does *not* necessarily contain the vertices of the polygon. If two points $p, q \in P$ are adjacent along the curve Σ (with no other point in P between them), then pq is a *correct* edge.

(a) We learned in class that if P is a 1-sample of Σ, then $\mathrm{DT}(P)$ contains every correct edge. Explain why this theorem is not useful in this context.

(b) Prove that if no two edges of the polygon meet at an angle less than 90°, and we replace the usual curve reconstruction definition of "local feature size" with Ruppert's definition, then if P is a 0.999-sample of Σ, $\mathrm{DT}(P)$ contains every correct edge.

[5] A three-dimensional BSP tree for point location (6 points). Let S be a set of n axis-aligned rectangular prisms in $\mathbb{R}^3$ with disjoint interiors. We wish to build a binary space partition tree to perform point location queries: given a point $p \in \mathbb{R}^3$, return a rectangular prism that contains p, or report that there is none. (If p lies on a prism boundary, it doesn't matter whether we treat it as being inside or outside the prism.) We choose a permutation of the prisms' $6n$ rectangular faces uniformly at random, then use that ordering of the faces to construct an autopartition. Show that for the BSP tree thus built,

(i) point location queries run in expected $O(\log n)$ time and

(ii) the expected size of the tree is in $O(n \log n)$. Hint: show that each rectangular 2-face is cut into expected $O(\log n)$ fragments. (The BSP complexity proof for segments from the Dutch Book shows that each segment in a randomized autopartition cuts expected $O(\log n)$ other segments, but I don't think that approach works here.)