CS 860, Fall 2004 due Monday October 18

A. Lubiw

ASSIGNMENT 1

1. Two exercises in geometric visualization:
   (a) Find a polyhedron that has convex faces but is not convex. What is the smallest such example?
   (b) Does the following procedure work to construct the dual of a convex polyhedron: Take a point in each face, such as the centroid (center of mass) of the face, and take the convex hull of these points.

2. Recall the divide-and-conquer algorithm for finding the convex hull of a set of points in 3 dimensions: separate the points in half by a plane; recursively compute the convex hull of each half, obtaining convex polyhedra A and B, say; and then find the convex hull of A and B. The portion of the boundary of A that disappears in the final step is the portion that is "visible" from B—i.e. the set of points a on the boundary of A for which there exists a point b on the boundary of B such that the line segment ab does not contain points interior to A or B. Prove that the "visible" faces of A form a connected set, but that invisible faces need not form a connected set. ("Connected" means in the dual graph, where two faces are adjacent if they share an edge.)

3. An intuitively obvious way to check if a polygon in the plane is convex is to check that the angle at each vertex is no more than π. Extend this to 3-dimensions: Give an algorithm to decide if a polyhedron is convex. You should clearly *state* (even if you don't prove) what is needed to justify the correctness of your algorithm. Your starting point may be the original definition of convex (that the line segment between any two points in the polyhedron remains inside the polyhedron) or the result that a polyhedron is convex iff it is the intersection of half-planes. You should also analyze the running time of your algorithm. You do not need to give low-level details of the computations.

4. Devise a simple algorithm to find a triangulation (not necessarily a Delaunay Triangulation) of a point set in 3-D. A "triangulation" means a set of tetrahedra with these properties:
   - two tetrahedra may interesect only at a common face or vertex
   - the tetrahedra partition the convex hull of the point set
   - the set of vertices of the tetrahedra is precisely the point set

   Analyze your algorithm. Is it optimal in the worst case?