Piyush Kumar

Computational Geometry, CIS 5930, Spring 2009

Assignment #2

Due: Jan 27th 2009

Each of the problems should be solved on a separate sheet of paper to facilitate grading. Staple all these sheets together! Please don't wait until the last minute to look at the problems.

Problem 1 [Source: David Mount]

The objective of this problem is to consider an alternative approach for an $O(n \log h)$ time algorithm for computing the upper hull of a set of points in the plane. Given two nonempty planar point sets A and B, which lie on opposite sides of a vertical line, the upper tangent of A and B is defined to be a nonvertical line T that passes through one point $a \in A$ and one point $b \in B$ such that no point of $A \cup B$ lies above this line. (See Figure 1.) It is a fact (which you need not prove) that the upper tangent exists and is unique.

Consider the following variation of the divide and conquer convex hull algorithm, which given an n-element planar point set P computes its upper hull $H(P)$.

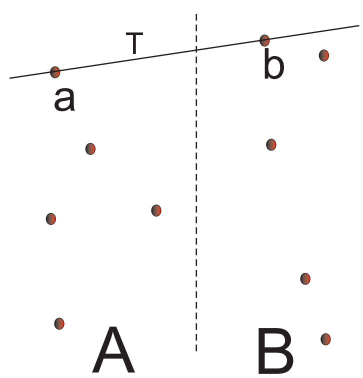

Figure 1: The upper hull of point sets A and B

1. If $|P| \leq 3$, then compute the upper convex hull by brute force in $O(1)$ time and return.

2. Otherwise, partition the point set P into two sets A and B, where A consists of $\lfloor n/2 \rfloor$ the points with the lowest x-coordinates and B consists of $\lceil n/2 \rceil$ of the points with the highest x-coordinates.

3. Compute the upper tangent T between A and B and let $a \in A$ and $b \in B$ be the contact point with these sets. (Note: This is done without knowing the upper hulls.)

4. Remove from A and B all the points whose x-coordinates lies strictly between a_x and b_x. (Do not remove a and b.) Let A' and B' be the respective subsets of remaining points.

5. Recursively compute the upper hulls $H(A')$ =`upperHull`(A') and $H(B')$ =`upperHull`(B').

6. Merge the two upper convex hulls $H(A')$ and $H(B')$ along with the segment ab to form the final upper hull.

The main difference between this algorithm and the divide-and-conquer algorithm given in class is that this algorithm computes the upper tangent before it computes the two hulls, rather than after. You may assume (without proving) that the computations of Steps (2) and (3) can be performed in $O(n)$ time, where $n = |P|$. (Step (2) follows from the well-known fast median algorithm and we will discuss Step (3) later this semester.) Using these facts, show that the algorithm described here takes $O(n \log h)$ time, where h is the number of points in the final upper convex hull.
Extra Credit : Implement this algorithm for your convex hull implementation and engineer it to be faster than Graham Scan or QuickHull at least for uniformly distributed points.

Problem 2
Given a set of points $P = \{p_1, p_2, \ldots, p_n\}$ in two dimensions, implement a class in C++ that outputs the Convex hull of this point cloud. It should be able to handle degeneracies of any kind. You will be using the predicates implemented in `predicates.c` from inside `dpoint.hpp` for the implementation. Your algorithm should run in $O(n \log n)$. You are encouraged to use `openmp` or `pthreads` to speed your implementation up. I will be running your implementations on a 8-core machine. For starters, download `assignment2.cpp, predicates.c` and the `Makefile` from the useful code page to get started. You should then download all the header files that you need (I usually use `wget` to do this). Once you have a compilable version, start modifying it.

Your code should be well documented using `Doxygen`.