COT5520: COMPUTATIONAL GEOMETRY

Homework # 2

Due date: Oct 21, 2008, Tuesday (beginning of the class)

Your solutions should be concise, but complete, and typed (or handwritten clearly). Feel free to consult textbooks, journal and conference papers and also each other, but write the solutions yourself and cite your sources. Answer **only five** of the following six questions. Each problem is worth 20 pts.

1. Consider a simpler, two-dimensional version of the casting problem discussed in class. Describe a linear time algorithm that decides whether a given polygon can be casted using a single mold and translation.

2. On n parallel railway tracks n trains are going with constant speeds $v_1, v_2, \ldots, v_n$. The trains are at positions $k_1, k_2, \ldots, k_n$ at time $t = 0$. Describe an $O(n \log n)$ time algorithm that reports all trains that at some moment in time are leading. To this end, use the algorithm for computing intersection of half-planes discussed in class.

3. Implement the kd-tree and the range tree data structures for performing two-dimensional rectangular range queries. Perform an experimental comparative study on data sets of various size (e.g., 100-100K) and of various distribution (e.g, random, convex position).

4. Kd-trees can also be used when querying with ranges other than rectangles, e.g., triangles.
   (a) Show that query time for range queries with triangles is linear in the worst case.
   (b) Suppose we limit our queries to triangles whose edges are horizontal, vertical or have slope +1 or -1. Design a linear size data structure that answers such range queries in $O(n^{3/4} + k)$ time, where k is the number of points reported.

5. *Range counting queries* ask for the number of points that lie in a range rather than the list of points. Describe how a d-dimensional range tree can be adapted such that a range counting query can be answered in $O(\log^d n)$ time.

6. Show how to combine the range tree and the priority search tree to obtain a data structure that answers a four-sided range query in the plane in time $O(\log n + k)$, without using fractional cascading.