

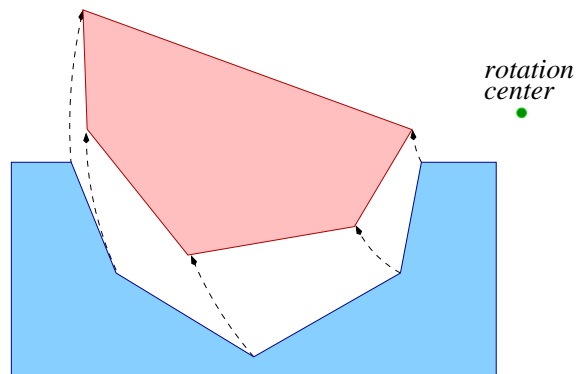
# COT5520: COMPUTATIONAL GEOMETRY

## Homework # 2

**Due date:** Oct 6, 2009, 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 seven questions. Each problem is worth 20 pts.

1. Instead of removing the object from its mold by a single translation, we can also try to remove it by a single rotation. For simplicity, let's study the planar version of the problem and consider clockwise rotations only.



- (a) Give an example of a simple polygon  $P$  with top facet  $f$  that is not castable when we require that  $P$  is removed from the mold by a single translation, but that is castable using rotation around a point. Also give an example of a simple polygon  $P$  with top facet  $f$  that is not castable when we require that  $P$  is removed from the mold by a rotation but that is castable using a single translation.
  - (b) Show that the problem of finding a center of rotation that allows us to remove  $P$  with a single rotation from its mold can be reduced to the problem of finding a point in the common intersection of a set of half-planes.
2. Implement an efficient algorithm that computes whether a given simple polygon is castable through a single translation (or rotation). The output should be visual showing the polygon, a top facet together with a translation vector (or a rotation center and direction).
  3. Consider the restricted version of the casting problem in which we insist that the object is removed from its mold using a vertical translation (perpendicular to the top facet.)
    - (a) Prove that in this case there is always a constant number of possible top facets.
    - (b) Give a linear time algorithm that determines whether for a given object a mold exists under this restricted model.
  4. In class we discussed a linear time algorithm for computing a random permutation. The algorithm needed a random number generator that can produce a random integer between 1 and  $n$  in constant time. Now assume we have a restricted random number generator available that can only generate a random bit (0 or 1) in constant time. How can we generate a random permutation with this restricted random number generator? What is the running time of your procedure?

5. 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).
6. Kd-trees can also be used when querying with ranges other than rectangles, e.g., triangles.
  - (a) Show that query time in a kd-tree of  $n$  points for range queries with triangles is  $\Omega(n)$  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.
7. One can use the data structure described in Chapter 5 to determine whether a particular point  $(a, b)$  is in a given set by performing a range query with range  $[a : a] \times [b : b]$ .
  - (a) Prove that performing such a range query on a kd-tree takes time  $O(\log n)$ .
  - (b) What is the time bound for such a query on a range tree? Prove your answer.