6.838: Geometric Computing

Spring 2005
Problem Set 4
Due: Thursday, May 12

Mandatory Part

Problem 1. External queue

Show how to efficiently implement a queue in external memory. That is, show how to support two operations:

- Push(a): pushes the element a at the *end* of the queue
- Pop: removes the element from the *front* of the queue and returns it

Your implementation should support any m operations on the queue using $O(m/B+1)$ block operations.

Problem 2. Metrics under translation

Consider any metric $D(A,B)$ defined for sets $A, B \subset \Re^2$. Assume that $D(\cdot,\cdot)$ is symmetric and satisfies triangle inequality. Let T be the set of all translations in $\Re^2$, and define $D_T(A,B) = \min_{t \in T} D(t(A), B)$. Show that $D_T(A,B)$ is symmetric and satisfies triangle inequality as well.

Problem 3. Deterministic packing.

In the lecture, we have seen a polynomial-time approximation scheme for the unit disk packing problem. The algorithm was randomized, since it used a randomly shifted grid.

Show that there is a *deterministic* polynomial-time approximation scheme for this problem.

Hint: The randomized algorithm uses a grid shifted by a vector chosen uniformly at random from $S = [0,k]^2$, where k is the side length of the grid cell. Thus, a deterministic algorithm could be obtained by trying out "all" vectors from S, and choosing the one that gives the best packing.

Optional Theoretical Part

Problem A. Reference point for EMD

Recall that, for two points sets $A, B \in \Re^2$, $|A| = |B|$

$$EMD(A,B) = \min_{\pi: A \to B} \sum_{a \in A} \|a - \pi(a)\|$$

where π is $1 - to - 1$.

Construct a reference point function $r(A)$ for EMD (under translations). That is, give a function $r(\cdot)$ such that for any two sets A, B, if $t^* = r(B) - r(A)$ is a translation that moves $r(A)$ to $r(B)$, then

$$EMD(t^*(A), B) \leq c \cdot \min_{t \in T} EMD(t(A), B)$$

for some constant $c \geq 1$. As before, T is the set of all translations in $\Re^2$.

Optional Programming Part

Problem B. Directed Hausdorff Under Translation

Implement a Java applet that simulates the algorithm from slide 13 of lecture 20. The algorithm should take as an input two sets of points A and B. Then it should do the following: if there is a translation t such that $DH(t(A), B) \leq r$, it should find a translation t' such that $D(t'(A), B) \leq (1 + \epsilon)r$. It is OK if ϵr is set to be equal to the diameter of a screen pixel (i.e., the grid used in the algorithm can coincide with the grid induced by screen pixels).

To implement the above, the algorithm should construct approximations of the sets $T(a)$ for all $a \in A$, and depict them on the screen. Then it should check if the intersections of all those sets is non-empty. If so, it should visualize the resulting translation.