Homework 1

Computer Graphics 2 (15-463)

6 Apr. 2001

Due 4pm Thursday, **12 Apr.** at my temporary office (NSH 4130) or in class that day.

Explain your work. You may use Maple, Mathematica, or other symbolic algebra systems for this homework, but if you do, turn in a transcript of your session.

1. [12 pts.] Let's define a *line segment split (LSS) BSP tree* to be the type of 2-D BSP tree described in the 462 lecture notes (refer to them if necessary): it is used to store a set of line segments, and its splitting lines are the lines coincident with line segments. When a line segment spans a splitting line, it is split into two line segments that are placed into their respective subtrees. Leaf cells have no line segments in their interior.

a) What are the shallowest and deepest LSS BSP trees possible with 7 disjoint (not intersecting, not touching), noncolinear line segments? Draw two views for each case: give both a geometric view of the line segments with the split lines passing through them, dividing the plane into regions (might be useful to use black & red here to distinguish line segments from split lines) and also a view of the BSP tree drawn as a tree with nodes and links. Label each line segment or node with a number or letter. See Foley's figure for an example of two such views.

b) What are the shallowest and deepest LSS trees possible with n disjoint, noncolinear line segments? Give formulas.

c) What is the worst case behavior of an LSS BSP tree with n possibly intersecting line segments? That is, what is the maximum number of nodes and what is the maximum depth, in $O()$ notation? Sketch an example. Hint: the max number of nodes involves lots of line segments getting split.

d) If we don't restrict ourself to LSS BSP trees, but allow arbitrary splitting lines, not just lines through a line segment in the set, how much can this improve the deepest tree from part **a**? That is, it we take the geometry yielding the deepest tree in part **a**, and build an unrestricted BSP tree for it, what is the shallowest this tree can be? Draw the two views, as before.

e) Now we jump to 3-D BSP trees defined for planar polygons in an analogous fashion to LSS BSP trees. What is the maximum number of nodes asymptotically? Describe a configuration yielding this number.

2. [12 pts] This problem analyzes uniform grid subdivisions for ray casting. Assume that there are s surfaces uniformly distributed in a box which is divided into an $n \times n \times n$ grid of rectangular voxels, and the voxels are bigger than the surfaces for the values of n of interest, so that each surface is listed in no more than 8 voxels' lists. Let the time to step from one voxel to its neighbor be t_s, and the time to intersect a ray with a surface be t_i.

a) What is the worst case time cost of intersecting a ray with the surfaces to find the first intersection point, as a function of n, s, t_s, and t_i. Your answer should have the units of time, and should not employ $O()$. For the worst case, keep the assumption that surfaces are uniformly distributed, (so they wouldn't all lie in one voxel), but find the ray path that will take the longest to trace. State your assumptions clearly.

b) Graph cost as a function of n, qualitatively (don't worry about constants for the graph, but show its character. Hint: the curve should be concave upwards and have a minimum, since intersection testing gets very expensive for n small, and stepping gets very expensive for n large.

c) Where is the minimum, and what is the time cost there?

d) Now, assuming we choose n optimally, graph the time cost as a function of s. For comparison, on the same graph, plot the cost of ray-surface intersection testing without a spatial data structure (this will be linear in s).