David Salesin, 418 Sieg
Head TA: Doug Zongker

Week 4 study sheet: Hidden Surface Algorithms and Ray Tracing

Problem 1 (12 points) An ideal hidden surface algorithm should:

i) quickly reject most of the hidden geometry in a model; and

ii) exploit the spatial coherence of the images being generated.

Evaluate each of the following hidden surface algorithms with respect to the two criteria above. (A single sentence saying why the algorithm does well or does not do well with respect to each criterion will do.)

- Z-buffer

  i)

  ii)

- Warnock's algorithm

  i)

  ii)

- BSP trees

  i)

  ii)

Problem 2 (21 points) The Z-buffer algorithm can be improved by using an image space "Z-pyramid." The basic idea of the Z-pyramid is to use the original Z-buffer as the finest level in the pyramid, and then combine four Z-values at each level into one Z-value at the next coarser level by choosing the farthest (largest) Z from the observer. Every entry in the pyramid therefore represents the farthest (largest) Z for a square area of the Z-buffer.

a) (2 points) At the coarsest level of the pyramid there is just a single Z value. What does that Z value represent?

Suppose we wish to test the visibility of a polygon P. Let Z_P be the nearest (smallest) Z value of polygon P. Let R be the smallest region in the Z-pyramid that completely covers polygon P, and let Z_R be the Z value that is associated with region R in the Z-pyramid.

b) (3 points) What can we conclude if $Z_R < Z_P$?

c) (3 points) What can we conclude if $Z_P < Z_R$?

Problem 2 (continued)

If the visibility test is inconclusive, then the algorithm applies the same test recursively: it goes to the next finer level of the pyramid, where the region R is divided into four quadrants, and attempts to prove that polygon P is hidden in each of the quadrants of R that P intersects. Since it is expensive to compute the closest Z value of P within each quadrant, the algorithm just uses the same Z_P (the nearest Z of the *entire* polygon) in making the comparison in every quadrant. If at the bottom of the pyramid the test is still inconclusive, the algorithm resorts to ordinary Z-buffered scan conversion to resolve visibility.

d) (6 points) Suppose that, instead of using the above algorithm, we decided to go to the expense of computing the closest Z value of P within each quadrant. Would it then be possible to always make a definitive conclusion about the visility of P within each pixel, without resorting to scan conversion? Why or why not?

Another type of coherence that one might like to exploit in a hidden surface algorithm is *temporal coherence*—the fact that the same polygons are likely to remain visible from one frame of an animation to the next.

e) (7 points) Suppose that we kept track of the polygons that were at least partially visible for some frame t. How might we use this information in frame $t+1$ to exploit temporal coherence with a Z-pyramid? How much better is using a Z-pyramid in this case than using a simple Z-buffer?

Problem 3 (6 points)

What is the asymptotic time complexity of ray tracing a scene consisting of m light sources and n objects, assuming that an image of p pixels is to be created and that rays are only traced to a recursive depth of at most 5?

Problem 4 (9 points) "Distributed ray tracing" can be used to simulate many real-world effects absent from ordinary recursive ray tracing. Name three such effects and describe briefly how distributed ray tracing can be used to achieve them.