Computer Science and Engineering 457

Introduction to Computer Graphics

Homework 2a

Spring Quarter, 2012; University of Washington

Due Wednesday, May 16 at the beginning of class

Instructions: Do this assignment individually (NOT in partnerships). Legibly write your answers on sheets of paper. When giving explanations, make sure they are clear. If there are corrections or clarifications to these problems, they will be posted on the course website, linked from the homepage. This is version 1.01.

1. Z-Buffers and Hierarchies (10 points)

   The computing time for z-buffer operation can sometimes be reduced by using a recursive subdivision of the image plane into quadrants and subquadrants, as needed. This method works by maintaining a tree structure known as a pyramid or balanced quadtree.

   Given a z-buffer of dimension n by n (where n is a power of 2) with contents $Z[i, j]$, construct a balanced quadtree as follows.

   Let $k' = \log_2 n$.
   Let $q[k', i, j]$ be a leaf having value $v(q[k', i, j]) = Z[i, j]$.
   For $k = k' - 1$ down to 0:
   $$\text{let } q[k, i, j] = \min(v(q[k+1, 2i, 2j]), v(q[k+1, 2i, 2j+1]),$$
   $$v(q[k+1, 2i+1, 2j]), v(q[k+1, 2i+1, 2j = 1]))$$

   Thus we define an internal node (a parent node) at level k for every four nodes at level $k+1$ and we give it a value that is the minimum (farthest away depth) of the values of those four children. Finally, the root is $q[0, 0, 0]$ and has the value $v(q[0, 0, 0])$ which is the minimum of all $Z[i, j]$. Note that each node of this quadtree represents a square region of the space covered by the original z-buffer. The region covered by a node can be denoted $r(q[k, i, j])$.

   Now suppose that a triangle T is to be tested with this z-buffer to determine whether any part of it lies in front of what's been processed so far. Let the closest (maximum) depth value of T be z_T. Let $q[k, i, j]$ be the node such that $r(q[k, i, j])$ is the smallest region of any quadtree node that completely covers the 2D projection of T.

   If $z_T < r(q[k, i, j])$ then the closest point of T is more distant than the most distant depth value covered by this node, and so T must be completely occluded. If not, then we can consider the four children of $q[k, i, j]$ to find out whether there is any possibility that part of T is visible within their regions. We can do this test quickly if we simply re-use z_T and do not try to determine a new z for the intersection of T with each child region.

   When we finally consider a leaf node in the tree, if z_T is closer than its v, then we determine the exact depth of T at the x and y values of i and j as usual with a z-buffer and update the z-value there if necessary. If the z-value changes, then we recompute the minimum among this node and its three siblings and if the minimum changes, we update the parent value, recompute the minimum with its children, etc.

   (a) (2 points) What is the significance of the value at the root of the quadtree?

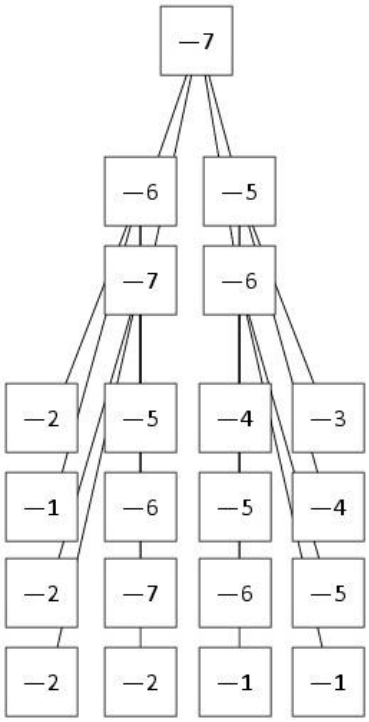


Figure 1: Quadtree representing a hierarchical organization of z-buffer contents.

(b) (2 points) If the quadtree's internal node values were computed using max instead of min, and no other changes made to the algorithm, how would the method behave differently?

(c) (2 points) Describe a situation for which about half of the z-buffer cells covered by a triangle T do not have to be touched by the algorithm. Assume that this T is entirely within the viewing frustum.

(d) (4 points) One form of aliasing occurs when the square region corresponding to a pixel in a z-buffer contains an intersection between two primitives (e.g., triangles), each of which should contribute some color to the pixel, but the pixel is shaded according to a sample from only one of them.

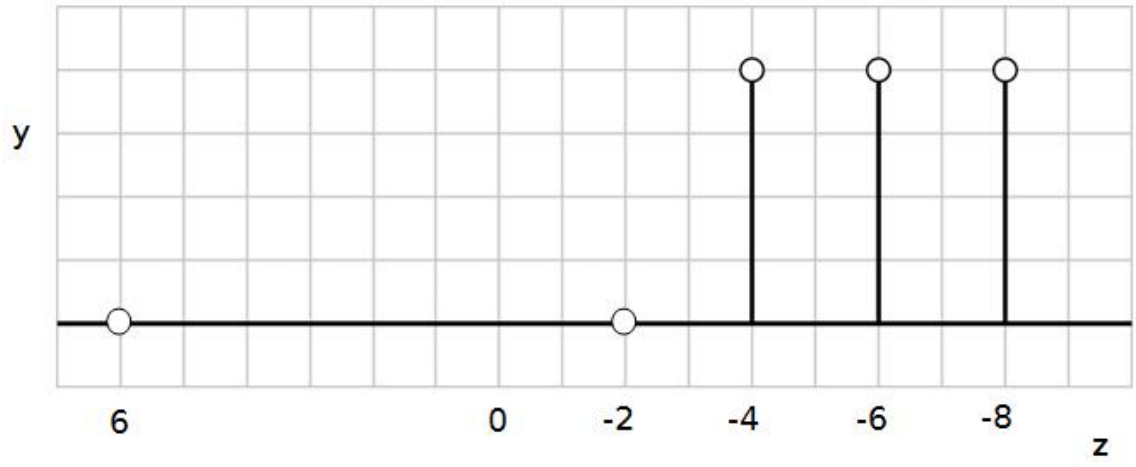
Describe a method for antialiasing that addresses this by using an extra level of nodes in the quadtree. (We assume that the display resolution does not change.) What additional storage is needed for the colors of subpixels, and what additional processing steps must be taken?

2. (10 points) The Dolly Zoom

   The cinematographic technique known as a dolly zoom was used in films such as *Vertigo* and *Jaws* to add an element of eeriness to a shot.

   Using the template below, diagram the projections of the three objects (line segments in this case) at the beginning and end of a dolly zoom subject to the following constraints.

   - The COP starts at $z = 6$.
   - The COP ends at $z = -2$.
   - The PP stays at $z = -3$.



(a)

(b)

Figure 2: Templates for your diagrams of the dolly-zoom planning. In the first, show the starting position projection, and in the second, show the ending position projection.

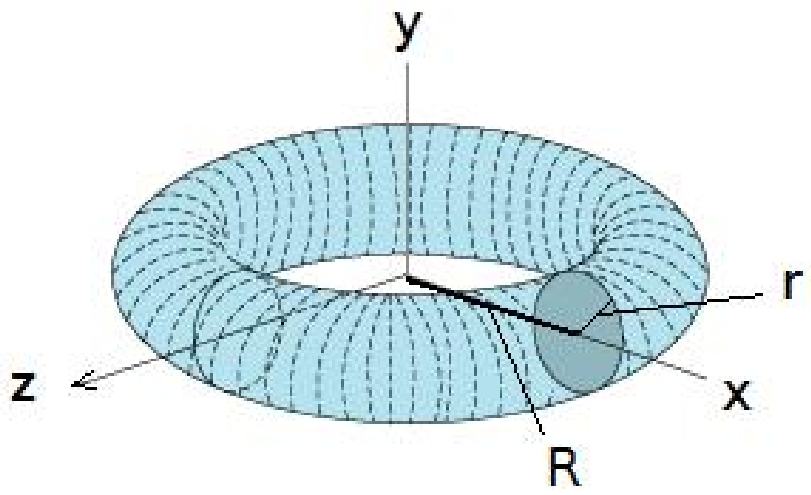


Figure 3: A torus centered at the origin.

3. (14 points) Intersecting Rays with Implicit Surfaces.

   One way to represent a curved surface is using an equation of the form $f(x, y, z) = 0$. This is known as an *implicit surface* representation. An implicit surface representation for a sphere is the following:

   $$x^2 + y^2 + z^2 - r^2 = 0$$

   Let us now consider a torus T, defined by the formula:

   $$(x^2 + y^2 + z^2 - r^2 - R^2)^2 + 4R^2(y^2 - r^2) = 0$$

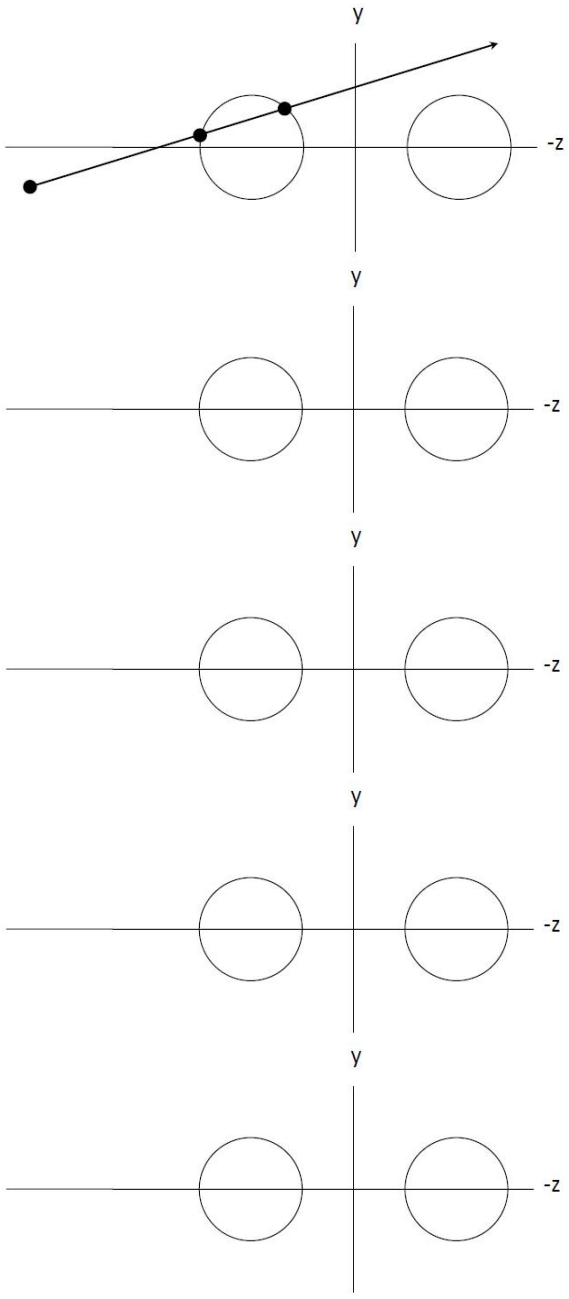
   Here r represents the inner radius of the torus and R represents its outer radius. For the remainder of this problem, you may assume that $r = 1$ and $R = 2$.

   In the following, you will be finding or illustrating intersections of rays with the torus. In general, the intersections are found by finding the roots of polynomials. In principle, the number of roots of a degree-n polynomial is n. Some of the roots may be complex (in which case its complex conjugate is also a root), and some of the roots may be duplicated (i.e., with multiplicity 2, 3, etc.). When a root is complex the corresponding intersection does not exist. Keep these ideas in mind.

   (a) (8 points) Consider a ray $P = P_0 + t\vec{d}$ starting at $P_0 = [0, 0, 4]$ (at the left of the torus in the figure) having direction $\vec{d} = [0, 0, -1]$. Find all values

of t where P intersects T. Which of these values of t is the important one for ray-tracing in this example, assuming the surface is opaque?

(b) (6 points) Explore the possible cases of ray intersections with the torus in terms of the number and types of intersections. Using the diagram templates below, show rays in the y-z plane that illustrate each case: 0 intersections, 1 intersection, etc. Please write on this page and turn it in with your homework solution. You do not need to justify your answers or show any formula or arithmetic. If there are any cases that cannot be effectively shown using these templates, then draw any additional diagram you need.


y
-z
y
-z
y
-z
y
-z
y
-z

4. (10 points) The Economics of Antialiasing in Ray Tracing

   In this problem, you are to give several expressions that represent the number of ray-primitive intersections that need to be determined under various assumptions about how the ray tracing is performed. In each case, assume that we are creating an n-by-n image of a scene involving N triangles. Your expressions should represent the worst-case numbers. (Additional assumptions apply to each situation.) Give closed-form solutions when possible. Otherwise, give summations with sigma.

   (a) (2 points) One viewing ray per pixel, ambient light only.

   (b) (2 points) One viewing ray per pixel, shadow rays for each of N_L point light sources, no reflection or refraction rays.

   (c) (2 points) One viewing ray per pixel, shadow rays as above, reflection and refraction rays to a maximum tree depth of d.

   (d) (2 points) Multiple viewing rays per pixel (k^2). Other rays as in the previous case.

   (e) (1 point) In order to obtain glossy reflection, in addition to the antialiasing of the previous case, you use k^2 rays for each original reflected ray.

   (f) (1 point) In order to obtain not only glossy reflection, as in the previous case, you also decide to implement translucent refraction by replacing each refracted ray by k^2 rays.