Intersect Algorithm

Algo Due 5pm Thursday, November 5, 2009
Help Session on Thursday, Novebmer 5, 2009 at 7:00pm

Name and Account:

1 Generating rays

For this assignment, you need to shoot a ray from the eye point through the center of each pixel.

1.1

Given a pixel with screen-space coordinates x and y, and the width and height of the screen x_{Max} and y_{Max}, what is the corresponding point, P_{film}, on the normalized film plane? Assume that this is taking place after all of the perspective viewing transformations have been applied except for the unhinging transformation. Use the far clip plane as your film plane, and remember that a pixel y-value of 0 corresponds to the top of the screen. (To be redundant, this point need not be defined in untransformed world space, as per question 1.2)

$P_{film} =$

1.2

You need to transform P_{film} on the normalized film plane into an untransformed world-space point, P_{world}. Using only the component matrixes of the viewing transformation listed below (or their inverses) write the equation for $M_{film\_to\_world}$, the composite transformation matrix that transforms P_{film} to P_{world}. Remember that the first matrix to be applied is listed last.

T, R, S_{xy}, S_{xyz}

$M_{film\_to\_world} =$

1.3

Given your eye-point P_{eye} and the world-space point on the normalized film plane P_{world} give the equation for the world-space ray you want to shoot into the scene. Specify your ray in the format $P + td$ where P is a point and d is a normalized vector.

$r(t) =$

2 Finding the closest object along each ray

2.1

Write out both of the cone-ray intersect equations in terms of t. Remember, there are two equations: one for the body of the cone, and one for the bottom cap. For your cone, use the same dimensions that you did in shapes. Use the defintion of a ray used in question 1.3.

Hint: To get you started you might want to define an intersection point as:

$(x, y, z) = (P_x + d_x t, P_y + d_y t, P_z + d_z t)$

Where P is the eyepoint, and $\vec{d}$ is the direction of the ray we are shooting. Looking over the the derivation of the implicit equations for the cylinder in the Raytracing lecture might also be useful.

The intersection points you compute are possible intersection points and need to be examined further (such as the $-0.5 \leq y \leq 0.5$ restriction for the body of the cylinder in the lecture notes). However for this problem you are NOT required to list these restrictions.

Note that in your program you will need to find intersection points by finding a value for t. If you do not find an explicit formula for t (ie. $t = somevalue(s)$) for both the cone and the cap then you will have a very hard time writing the program and you will also get **very little credit** for this problem. Finally the equations you write should not use vectors but should be functions of the individual components of the vectors.

3 Illuminating Samples

3.1

When you are attempting to illuminate a transformed object, you will need to know that object's normal vector in world-space. Assume you know the normal vector in object-space, N_{object}. Give an equation for the normal vector in world-space, N_{world}, using the object's modeling transformation M and N_{object}.

$N_{world} =$

3.2

In the lighting equation, what does $\vec{N} \cdot \vec{L}$ represent, i.e. what is equivalent to it? What is its purpose in the equation?