Introduction to Raytracing

1 Overview

Raytracers operate by emitting a finite set of light rays from the camera through a pixel grid. Each ray is emminated from the camera origin through a pixel center, checked for contact against any geometry in the scene, a lighting calculation is performed, and the pixel the ray passes through is given the color output of some lighting calculation. This process is repeated for every pixel in the grid. Because the raytracer simulates the path of light, interesting effects such as refraction and scattering are possible to do easily and accurately. The downside of raytracing is that it is currently not hardware accelerated, and must be done slowly in software.

2 Emitting Rays

Before we can begin emitting rays, we first need to establish the location of the camera origin, the distance of the pixel grid from the camera origin, and the dimensions of the grid. Let s denote the position of the camera origin, which we'll assume to be the zero vector. We'll also assume that the pixel grid is the same dimensions of the render window, $w \times h$ ($w = 640$, $h = 480$ by default), aligned with the X-Y axes, and z units down the negative-Z axis. The coordinates of integer-valued grid cell center (i, j), where $0 \leq i < w$ and $0 \leq j < h$, is then, $(i - w/2 + 0.5, j - h/2 + 0.5, -z)$.

That natural question is how is z calculated. In general, z, is determined by the field of view of the camera, θ. A narrow field of view makes objects appear further away, while a wider field of view makes them closer. For this assignment you should assume $\theta = \pi/2$ (45 degrees).

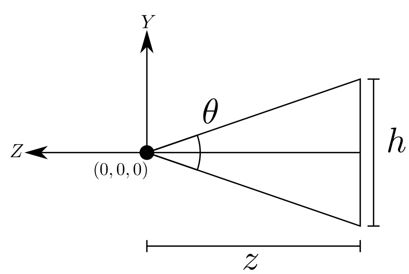


Figure 1: A side view of the camera frustum.

Given the field of view (see Fig. 1, we can determine the distance of the pixel grid from the camera origin from the following relationship:

$$\tan(\theta/2) = \frac{h}{2z} \ , \tag{1}$$

which, when rearranged for z, gives

$$z = \frac{h}{2\tan(\theta/2)} \ . \tag{2}$$

Once the grid cell point, x, is determined we can define the direction vector d as

$$d = x - s \ , \tag{3}$$

and use it to construct a parametric ray function which emanates from the camera origin:

$$
\begin{aligned}
\mathbf{P}(t) &= \mathbf{s} + t\mathbf{d} && (4)\\
&= \begin{bmatrix} s_x + td_x \\ s_y + td_y \\ s_z + td_z \end{bmatrix}, && (5)
\end{aligned}
$$

where t is the parametric variable which controls the scale of d.

3 Intersection Testing with Implicit Spheres

The implicit equation for a sphere centered at (c_x, c_y, c_z) is given by:

$$(x - c_x)^2 + (y - c_y)^2 + (z - c_x)^2 - r^2 = 0\,. \tag{6}$$

The point of intersection between the ray and the implicit sphere can be found by plugging Eqn. 5 into Eqn. 6 for x, y, and z,

$$(s_x + td_x - c_x)^2 + (s_y + td_y - c_y)^2 + (s_z + td_z - c_z)^2 - r^2 = 0\,, \tag{7}$$

and solving for t. Eqn. 7 can be solved using the Quadratic Equation:

$$
\begin{aligned}
t_1 &= \frac{-b + \sqrt{b^2 - 4ac}}{2a} && (8)\\
t_2 &= \frac{-b - \sqrt{b^2 - 4ac}}{2a} && (9)
\end{aligned}
$$

where

$$
\begin{aligned}
a &= d_x^2 + d_y^2 + d_z^2 && (10)\\
b &= 2(d_x(s_x - c_x) + d_y(s_y - c_y) + d_z(s_z - c_z)) && (11)\\
c &= (s_x - c_x)^2 + (s_y - c_y)^2 + (s_z - c_z)^2 - r^2 && (12)
\end{aligned}
$$

All quadratic equations have two real or imaginary solutions, t_1, t_2. In our problem the solutions (roots) correspond to the intersection points of the ray with the boundary of the sphere volume. There are three possible intersection cases that we must consider:

1. The ray intersects the sphere twice: two distinct real roots.
2. The ray grazes the sphere: two equal real roots.
3. The ray misses the sphere: two distinct imaginary roots.

We can detect these cases by examining the term inside the square root of Eqn. 9. If $b^2 - 4ac > 0$, then there are two distinct real valued roots and we end up with the first case above. Similarly, if $b^2 - 4ac = 0$ or $b^2 - 4ac < 0$, we end up with the second and third case, respectively.

If we have determined that there exists real valued roots, we need to identify the smallest non-negative value of t_1, t_2. The smallest non-negative value corresponds to the intersection point closest to the viewer that is not behind the viewer. If $t_1 < 0$ and $t_2 < 0$, we do not light the pixel.

Let t^* denote the smalled non-negative value. We can compute the point of intersection, p, from t^*:

$$p = s + t^* d\ . \tag{13}$$

Fig. 2 is an example of a raytraced sphere without the complete lighting calculations. Each pixel is set to black if the corresponding ray intersects the sphere ($b^2 - 4ac \geq 0$).

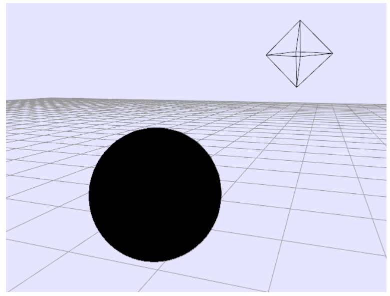


Figure 2: The sphere prior to the lighting calculations.

4 Lighting

Now that p has been determined, we can perform the necessary lighting calculations for the pixel. The color output by the Phong shader, C, is given by the formula:

$$C = l_a C_a + l_d C_d (N \cdot L) + l_s C_s (V \cdot R)^{\alpha} \tag{14}$$

where the three terms correspond to the ambient, diffuse and specular components, respectively; l_a, l_d, and l_s are the scalar ambient, diffuse, and specular light intensities; C_a, C_d, and C_s are the ambient, diffuse and specular color material vectors; α is the shininess coefficient; and N, L, V, R are the normalized normal, light, viewer, and reflectance vectors. Fig. 4 illustrates the effect of the various lighting components.

N, L, V, and R can be calculated as follows:

$$N = \frac{p - c}{\|p - c\|} \tag{15}$$

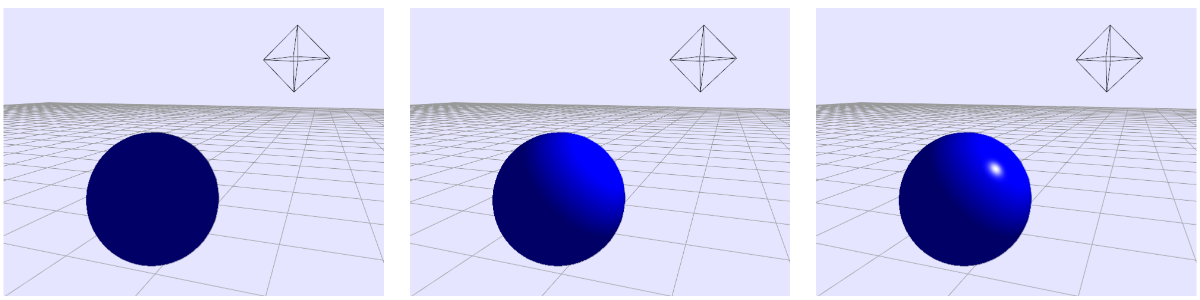
$$L = \frac{q - p}{\|q - p\|} \tag{16}$$

$$V = \frac{o - p}{\|o - p\|} \tag{17}$$

$$R = 2(N \cdot L)N - L \,, \tag{18}$$

where c, q, and o are the center of the implicit sphere, the position of the light, and the position of the camera origin, respectively.

You will also need to ensure that $N \cdot L \geq 0$ and $V \cdot N > 0$ before adding the diffuse and specular components (recall that the dot product is positive when the angle between the two normalized input vectors is less than 90 degrees). If this statement is not true, it implies that the light is occluded by the geometry. Similarly, if $R \cdot V < 0$ we do not add the specular component because the light is bouncing away from the viewers eye.



(a) ambient (b) ambient and diffuse (c) ambient, diffuse and specular

Figure 3: A Phong-shaded sphere with various lighting components active.