Ray-Casting Formulas

Ray - Sphere Intersection

A sphere is given by its center (cx, cy, cz), its radius R, and its color (SR, SG, SB).
A line segment (ray) is given by its endpoints: P0 = (x0, y0, z0) and P1 = (x1, y1, z1).
To find visible spheres, set P0 = viewer's coordinates, VP = (VPx, VPy, VPz) and let P1 run through all the points (x1, y1, 0) where (x1, y1) is a pixel in the display area.

Set dx = x1 - x0
dy = y1 - y0
dz = z1 - z0

a = dx*dx + dy*dy + dz*dz;
b = 2*dx*(x0-cx) + 2*dy*(y0-cy) + 2*dz*(z0-cz);
c = cx*cx + cy*cy + cz*cz + x0*x0 + y0*y0 + z0*z0 +
-2*(cx*x0 + cy*y0 + cz*z0) - R*R;

discriminant(a, b, c) = b^2 – 4*a*c

if discriminant (a, b, c)		
	< 0	no intersection
	= 0	ray is tangent to sphere
	> 0	ray intersects sphere in two points

The intersection nearest P0 is given by:

t = (-b – sqrt(b^2 – 4*a*c))/(2*a)

To find the coordinates of the intersection point:

x = x0 + t*dx y = y0 + t*dy z = z0 + t*dz

Ray-Casting Steps

For each pixel in your image

1. Find the visible surface

Use the Ray-Sphere Intersection formulas with

P0 = View Point = VP = (VPx, VPy, VPz)

P1 = pixel = (x1, y1, 0)

Intersect this ray with each sphere in your scene

If no sphere intersects set the pixel to the background color.

Otherwise set the pixel to the color of the sphere that gave the intersection with the smallest t.

Cp = (SR, SG, SB).

If you do only this, your picture should look like a number of overlapping disks on the plane.

2. Add Diffuse Shading

Do step ***1*** but replace the "Otherwise" with this:

Otherwise

Find the coordinates of the point (x, y, z) on the sphere.

x = x0 + t*dx y = y0 + t*dy z = z0 + t*dz

Find the unit normal vector to the sphere at (x, y, z)

N = ((x - cx)/R, (y - cy)/R, (z - cz)/R)

Find a unit vector from (x, y, z) to the light (Lx, Ly, Lz).

L = (Lx - x, Ly - y, Lz - z)/|| (Lx - x, Ly - y, Lz - z)||.

Compute fctr = N•L = cos of the angle between N and L.

Set the pixel to

Cp = (ka*SR, ka*SG, ka*SB) + (kd*fctr*SR, kd*fctr*SG, kd*fctr*SB),

where kd is the diffuse-coefficient and ka is the ambient-coefficient. (Try kd = .8 and ka = .2 or experiment with your own values.)

3. Find the Shadows

If the pixel is background color, use the Ray-Sphere Intersection formulas with

P0 = pixel = (x0, y0, 0)

P1 = Light = (Lx, Ly, Lz)

Intersect this ray with each sphere in your scene.

If there is any intersection, the pixel is in shadow.

Use half (or less) the R, G, B of your background color.

If there is no intersection, use the full R, G, B of the background color.

If the pixel is about to be colored to show a sphere, use the Ray-Sphere Intersection formulas with

P0 = Point on sphere = (x, y, z)

P1 = Light = (Lx, Ly, Lz)

Intersect this ray with **every other sphere** in your scene.

If there is any intersection, the point is in shadow. Use just the ambient color of the visible sphere.*

Cp = (ka*SR, ka*SG, Ka*SB).

If there is no intersection, use the color computed in part ***2***.

Cp = (ka*SR, ka*SG, Ka*SB) + (kd*fctr*SR, kd*fctr*SG, kd*fctr*SB).

*Though it is not as physically correct, I sometime use

Cp = (ka*SR, ka*SG, Ka*SB) + (.5*kd*fctr*SR, .5*kd*fctr*SG, .5*kd*fctr*SB)

for points on the spheres that are in shadow. This makes the sphere look more rounded.

4. Add the Phong Highlights

If the pixel is about to be colored to show a point (x, y, z) on a sphere that is shiny (ks > 0) and the point is not in a shadow, add the Phong highlight.

L = (Lx, Ly, Lz) is a unit vector from the point on the sphere to the light (see ***2*** above).

V = (Vx, Vy, Vz) is a unit vector from the point (x, y, z) to the viewpoint VP.

$V = (Vx - x, Vy - y, Vz - z)/\|(Vx - x, Vy - y, Vz - z)\|$

H = (Hx, Hy, Hz) = (L + V)/||L + V|| is the *halfway vector*.

$Cp = ka(SR, SG, SB) + kd\ \mathbf{N}\bullet\mathbf{L}\ (SR, SG, SB) + ks\ (\mathbf{H}\bullet\mathbf{N})^{n}\ (1, 1, 1)$

Set hf = highlight factor = $(H\bullet N)^{n}$.

Cp = (ka*SR, ka*SG, Ka*SB) + (kd*fctr*SR, kd*fctr*SG, kd*fctr*SB) + (ks*hf , ks*hf , ks*hf).