1.1 We do our primary ray generation in world space. We shouldn't do this in screen space because though our eye starts at a screen coordinate pixel, it must go "through" the scene to see which object it hit, into the actual "world". We shouldn't do this in object space because the illumination model prefers the ray-object intersection point to be in world-space. This is because it is less work than normalizing the lights with geometry.

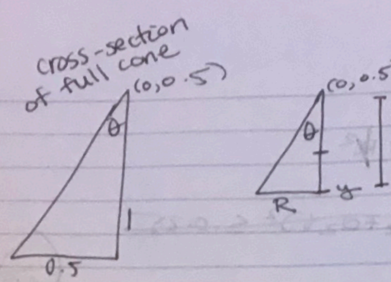
1.2 (2*Px/Xmax -1, 1- 2 * Py/Ymax, -1)
1.3 We need to apply the inverse of the normalizing transformation, the viewing transformation. We need to do this to transform to pre-normalized world space. This is because the illumination model prefers the intersection point between a vector and object to be in world space, as it is less work than normalizing lights and geometry. We multiply this matrix by a pixel screen coordinate to determine the world space equation for a ray that goes through that pixel screen coordinate. (MscaleMrotMtrans)^-1

1.4 Peye + t(Pworld-Peye)

2. (next page).

3.1 nWorld = $((M_3)^T)^{-1}$ nObj
3.2 nL is the angle of the incoming light. L is the unit vector to light, n is the unit surface normal. This ends up being equivalent to cos(theta) where theta is the angle to normal. As the angle increases, the light's energy spreads across a larger area.

4.1 Lighting is the process of computing the intensity and color of a sample point in a scene as seen by a viewer. It's a function of the geometry of scene and material properties. Shading is the process of interpolation of color at points in between those with known lighting or illumination, typically vertices of triangles in a mesh. Lighting is calculated by a vertex shader, while shading is done a by fragment or pixel shader.



by similar triangles:

$$\frac{0.5}{1} = \frac{R}{0.5-y}$$

$$R = 0.5\,(0.5-y)$$

$$x^2 + z^2 = R^2$$

$$(p_x + \vec{d}_x t)^2 + (p_z + \vec{d}_z t)^2 = (0.25 - 0.5y)^2$$

$$p_x^2 + 2\vec{d}_x t + (\vec{d}_x)^2 t^2 + p_z^2 + 2 p_z \vec{d}_z t + \vec{d}_z^{\,2} t^2 = 0.25^2 - (0.25)(2)(0.5y) + 0.25y^2$$

$$(d_x^2 + d_z^2)t^2 + (2d_x p_x + 2p_z d_z)t + (p_x^2 + p_z^2) = 0.25^2 - 0.25(p_y + d_y t) + 0.25(p_y + d_y t)^2$$

$$(d_x^2 + d_z^2 - 0.25d_y)t^2$$
$$+ (2d_x p_x + 2p_z d_z + 0.25d_y - 0.5d_y p_y)t$$
$$+ (p_x^2 + p_y^2 - 0.25p_y^2 - 0.25^2 + 0.25p_y) = 0$$

So, let

$$a = d_x^2 + d_z^2 - 0.25d_y$$
$$b = 2d_x p_x + 2d_z p_z + 0.25d_y - 0.5d_y p_y$$
$$c = p_x^2 + p_y^2 - 0.25p_y^2 - 0.25^2 + 0.25p_y$$

Then

$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

and we choose the smallest non-negative t-value

Gnarly. The cap's on the next page!

Cap $y = P_y + d_y t$

0.5 $-0.5 = P_y + d_y t$

$$\boxed{\frac{-0.5 - P_y}{d_y} = t}$$