Angel: Interactive Computer Graphics, Fourth Edition

Chapter 6 Solutions

6.1 Point sources produce a very harsh lighting. Such images are characterized by abrupt transitions between light and dark. The ambient light in a real scene is dependent on *both* the lights on the scene and the reflectivity properties of the objects in the scene, something that cannot be computed correctly with OpenGL. The Phong reflection term is not physically correct; the reflection term in the modified Phong model is even further from being physically correct.

6.3 If we were to take into account a light source being obscured by an object, we would have to have all polygons available so as to test for this condition. Such a global calculation is incompatible with the pipeline model that assumes we can shade each polygon independently of all other polygons as it flows through the pipeline.

6.5 Materials absorb light from sources. Thus, a surface that appears red under white light appears so because the surface absorbs all wavelengths of light except in the red range—a subtractive process. To be compatible with such a model, we should use surface absorbtion constants that define the materials for cyan, magenta and yellow, rather than red, green and blue.

6.7 Let ψ be the angle between the normal and the halfway vector, ϕ be the angle between the viewer and the reflection angle, and θ be the angle between the normal and the light source. If all the vectors lie in the same plane, the angle between the light source and the viewer can be computer either as $\phi + 2\theta$ or as $2(\theta + \psi)$. Setting the two equal, we find $\phi = 2\psi$. If the vectors are not coplanar then $\phi < 2\psi$.

6.13 Without loss of generality, we can consider the problem in two dimensions. Suppose that the first material has a velocity of light of v_1 and the second material has a light velocity of v_2. Furthermore, assume that the axis $y = 0$ separates the two materials.

Place a point light source at $(0, h)$ where $h > 0$ and a viewer at (x, y) where $y < 0$. Light will travel in a straight line from the source to a point $(t, 0)$ where it will leave the first material and enter the second. It will then travel from this point in a straight line to (x, y). We must find the t that minimizes the time travelled.
Using some simple trigonometry, we find the line from the source to $(t, 0)$ has length $l_1 = \sqrt{h^2 + t^2}$ and the line from there to the viewer has length

$l_2 = \sqrt{y^2 + (x-t)^2}$. The total time light travels is thus $\frac{l_1}{v_1} + \frac{l_2}{v_2}$. Minimizing over t gives desired result when we note the two desired sines are $\sin\theta_1 = \frac{h}{\sqrt{h}^2+t^2}$ and $\sin\theta_2 = \frac{-y}{\sqrt{(y^2+(x-t)^2}}$.

6.19 Shading requires that when we transform normals and points, we maintain the angle between them or equivalently have the dot product $\mathbf{p} \cdot \mathbf{v} = \mathbf{p}' \cdot \mathbf{v}'$ when $\mathbf{p}' = \mathbf{Mp}$ and $\mathbf{n}' = \mathbf{Mp}$. If $\mathbf{M}^T\mathbf{M}$ is an identity matrix angles are preserved. Such a matrix ($\mathbf{M}^{-1} = \mathbf{M}^T$) is called orthogonal. Rotations and translations are orthogonal but scaling and shear are not.

6.21 Probably the easiest approach to this problem is to rotate the given plane to plane $z = 0$ and rotate the light source and objects in the same way. Now we have the same problem we have solved and can rotate everything back at the end.

6.23 A global rendering approach would generate all shadows correctly. In a global renderer, as each point is shaded, a calculation is done to see which light sources shine on it. The projection approach assumes that we can project each polygon onto all other polygons. If the shadow of a given polygon projects onto multiple polygons, we could not compute these shadow polygons very easily. In addition, we have not accounted for the different shades we might see if there were intersecting shadows from multiple light sources.