

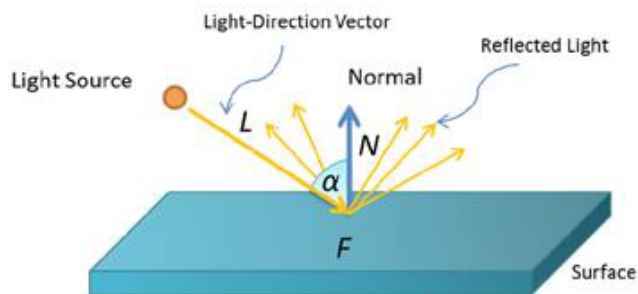
Implementing Diffuse Reflection in WebGL

Files: 07-lighting01.html, 07-lighting02.html, 07-lighting03.html, 07-lighting04.html, 07-lighting05.html

Recall that the formula for computing the surface color by diffuse reflection is given by the lambertian light reflection model which is

$$\text{DiffuseColor} = \text{LightDiffuseColor} \times \text{Material DiffuseColor} \times \cos \alpha$$

Lambertian Reflectance

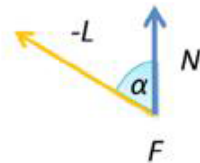


Final Diffuse Color

$$F = C_l C_m (-L \cdot N)$$

Light Diffuse Color Material Diffuse Color

Final diffuse color calculation for fragment F



$$-L \cdot N = |-L||N| \cos \alpha$$

If L and N are normalized then:

$$-L \cdot N = \cos \alpha$$

$$F = C_l C_m \cos \alpha$$

A Lambertian surface reflects light in many directions

Computing Correct Normals on Translated/Rotated Objects

Files: 07-lighting06.html, 07-lighting07.html

To correct the normals of translated/rotated objects, another matrix to represent the correct transformation must be used. This is usually called the normal matrix.

$$\textit{correctedNormal} = \textit{normalMatrix} * \textit{normalVector}$$

Normal Matrix is dependent on the model matrix that was used to transform the vertex position of the objects. The normal Matrix is computed as

$$(\textit{modelMatrix}^{-1})^T$$

or

$$\textit{normalMatrix} = \textit{transpose}(\textit{inverse}(\textit{modelMatrix}))$$

Exercise

Create a lighted cube (Lambertian LRM, Flat Shading, directional light)

