Table of Contents

Shading and Lighting Models

A shading model dictates how incident light is scattered or reflected from a surface. For simplicity, we assume achromatic light (adding colors is a simple extension). There are two types of light sources that illuminate objects: ambient light and point light sources. Incident light interacts with a given surface in the following ways:

- A portion of the incident light is converted to heat
- Another portion is reflected by the surface
- What remains traverses the surface (as with glass)

There are two ways in which light is reflected by a surface. Some of the incident light is diffusely scattered while the rest reflects in a specular (mirror-like) fashion. Most surfaces reflect incident light as a combination of these two models.

In general, three vectors (and the angles between them) are needed to model surface reflections. Given a point on a surface, a normal vector to the surface at the point, a vector from the point to the light source, and a vector from the surface point to the viewing point are needed.

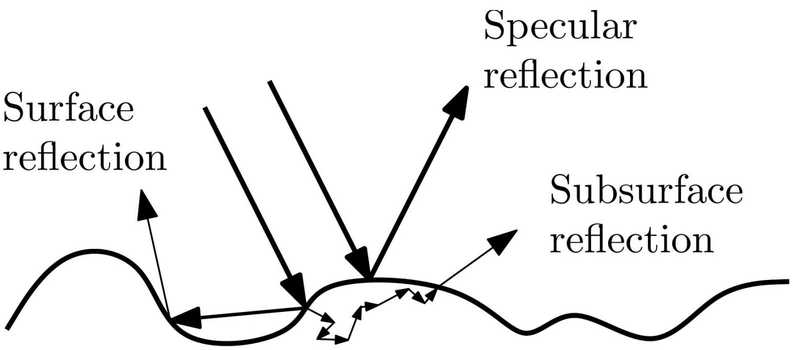

Illustration 1: The different ways incident light is reflected by a surface

The Diffuse Component of Light Reflection

When incident light reaches a surface, some of it is radiated diffusely and reaches the eye of the observer with intensity I_d . In order to compute this quantity, one needs to realize that diffuse scattering is independent from the direction from the surface point to the eye of the observer, as scattering is omnidirectional. Hence, the orientation of the surface with respect to the the eye of the viewer is not relevant in computing I_d . However, the amount of incident light the surface receives depends on its orientation relative to the light source.

The equation that allows us to compute I_d is thus

- independent from the angle of the surface normal $\vec{n}$ at point P and the vector $\vec{v}$ from P to the origin of the viewing coordinate system (the viewer's eye, or the focal point of the camera).
- Dependent on the angle between the vector $\vec{s}$ from P to the light source and the surface normal vector $\vec{n}$ at P .

The amount of diffusely reflected light that reaches the eye of the observer can thus be approximated as:

$$I_d = I_s \rho_d \left(\frac{\vec{s} \cdot \vec{n}}{\|\vec{s}\| \|\vec{n}\|} \right)$$

where I_s is the intensity of the light source, ρ_s is the diffuse reflection coefficient, and

$$\left(\frac{\vec{s} \cdot \vec{n}}{\|\vec{s}\| \|\vec{n}\|} \right) = \cos\theta$$

is the cosine of the angle between $\vec{s}$ and $\vec{n}$ .If the surface is aimed away from the eye (a condition determined by the direction of the normal with respect to the viewer), then the dot product is negative and we set I_d to zero. Hence, in order to be accurate, one should write the following:

$$I_d = I_s \rho_d \, max\left(0, \frac{\vec{s} \cdot \vec{n}}{\|\vec{s}\| \|\vec{n}\|}\right)$$

The Specular Component of Light Reflection

Modeling the specular component of light reflection is similar to that of the diffuse component, a few important details excepted. Given a surface point P , its normal vector $\vec{n}$, and a vector $\vec{s}$ from P to the light source, then the direction of perfect specular reflection is symmetrical to $\vec{s}$ with respect to $\vec{n}$ and is defined as:

$$\vec{r} = -\vec{s} + 2 \frac{\vec{s} \cdot \vec{n}}{\|\vec{n}\|^2} \vec{n}$$

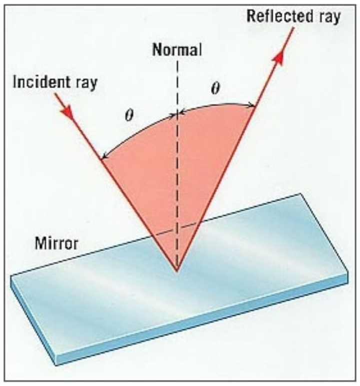

Illustration 2: The perfect specular reflection

Of course, only a perfect mirror reflects light in this way, whereas most surfaces reflect incident light in more than just this direction. As a consequence, the viewer's position determines the amount of light from specular reflection that reaches the eye. We can approximate this quantity with the following equation:

$$I_p = I_s \rho_p \, max\left(0, \frac{\vec{r} \cdot \vec{v}}{\|\vec{r}\| \|\vec{v}\|}\right)^f$$

where, this time, ρ_p is the specular reflection coefficient, $\vec{v}$ is the vector from the surface point to the eye of the viewer, and f is a fallout exponent governing how much specular light reaches the eye as $\vec{v}$ moves away from $\vec{r}$.

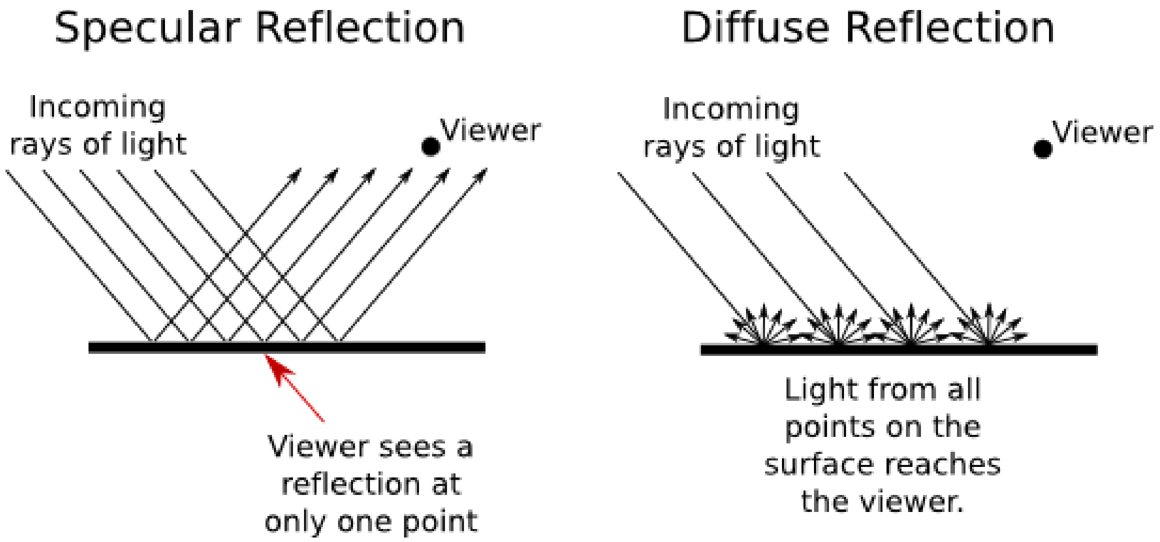
Specular Reflection
Incoming rays of light
Viewer
Viewer sees a reflection at only one point
Diffuse Reflection
Incoming rays of light
Viewer
Light from all points on the surface reaches the viewer.

Illustration 3: The difference between specular and diffuse reflection from the perspective of the viewer

Ambient Light

In any realistic scene, there are so many surfaces that reflect incident light coming from a source that it creates what is generally called ambient light. This type of light is computationally expensive to calculate and we prefer to simply associate an amount of ambient light to the rendered scene. In addition, without it, shadowed areas of objects would seem totally dark, which is in most cases unrealistic in terms of appearance. We associate an amount of ambient light of a scene (or a face of an object, if needed) simply as a light source with intensity I_a which has no position, and an ambient coefficient ρ_a , to form the component of ambient light simply as $I_a \rho_a$.

Combining Light Contributions

The total light intensity reaching the eye of the viewer receives from a surface point can thus be computed by adding the ambient, diffuse, and specular components in the following way:

$$I = I_a \rho_a + I_d + I_p$$

Smooth Shading

There are two well known smooth shading techniques: Gouraud shading and Phong shading. Gouraud shading involves the interpolation of intensity values while Phong shading works with the interpolation of surface normals.

Gouraud shading uses linear intensity interpolation and is a line scan method. Suppose a planar face of an object is projected onto the image plane of the viewing camera. Suppose also that the normals at the vertices are known as an average of the normals of the faces they belong to. In this case intensity can be interpolated from one vertex to the next in a linear way. Hence, at the intersection points formed by the scan line entering the projected face from the right and exiting it from the left, and with intensities known at these points, we can also interpolate the intensity values along the scan line in between the intersections. For example, suppose a projected face from an object with the following vertices: $P_1(x_{1,} y_1)=(1,4)$, $P_2(x_{2,} y_2)=(4,1)$, and $P_3(x_{3,} y_3)=(4,7)$ for which the normals at these points are known: $\vec{n}_1$, $\vec{n}_2$, $\vec{n}_3$. With these elements we can compute the intensities (as above) at the vertices of the face as I_1 , I_2 , and I_3 . Let's also suppose that the current horizontal scan line is at $y=3$. This line crosses the segments $P_1 P_2$ at $(2,3)$ and $P_2 P_3$ at $(4,3)$, and let's label these intersection points $P_4(x_{4,} y_4)=(2,3)$ and $P_6(x_{6,} y_6)=(4,3)$ respectively. At this point, with linear interpolation, we can obtain the intensities I_4 and I_6 as:

$$\begin{aligned} I_4 &= \left(\frac{y_4-y_2}{y_1-y_2}\right) I_1 + \left(\frac{y_1-y_4}{y_1-y_2}\right) I_2 \\ I_6 &= \left(\frac{y_6-y_2}{y_3-y_2}\right) I_3 + \left(\frac{y_3-y_6}{y_3-y_2}\right) I_2 \end{aligned}$$

For any point $P_5(x_{5,} y_5)$ on the scan line, we can now interpolate the correct intensity simply as:

$$I_5 = \left(\frac{x_6-x_5}{x_6-x_4}\right) I_4 + \left(\frac{x_5-x_4}{x_6-x_4}\right) I_6$$

Gouraud shading can be summarized as follows:

- Given a face from an object, compute the normal vector of each of its vertices by averaging the face normals for which the vertex is a part of
- Compute intensity values at the face vertices
- With the projected face vertices and their respective intensity values, use a scan-line approach to fill in the face with interpolated intensities

Alternatively, one may want to use a different method to perform smooth shading. Phong shading operates in a way similar to Gouraud, except that normal vectors are interpolated instead of intensities, and the formulae simply become (using the same example as above):

$$\begin{aligned} \vec{n}_4 &= \left(\frac{y_4-y_2}{y_1-y_2}\right) \vec{n}_1 + \left(\frac{y_1-y_4}{y_1-y_2}\right) \vec{n}_2 \\ \vec{n}_6 &= \left(\frac{y_6-y_2}{y_3-y_2}\right) \vec{n}_3 + \left(\frac{y_3-y_6}{y_3-y_2}\right) \vec{n}_2 \end{aligned}$$

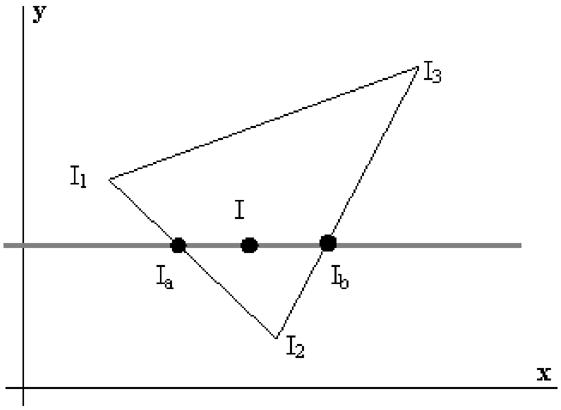

Illustration 4: Computing intensity at I with Gouraud shading

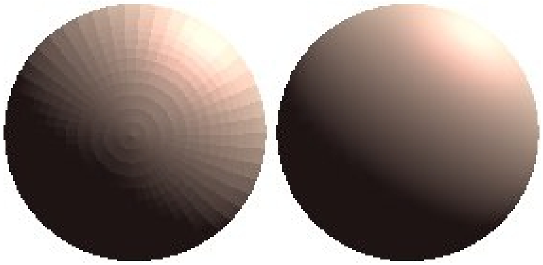

Illustration 5: The difference between a flat shading without interpolation and Gouraud shading of a sphere

and the normal at any point $P_5(x_{5,} y_5)$ on the scan line is obtained as:

$$\vec{n}_5 = \left(\frac{x_6 - x_5}{x_6 - x_4}\right)\vec{n}_4 + \left(\frac{x_5 - x_4}{x_6 - x_4}\right)\vec{n}_6$$

The intensities for the scan-line points are obtained as before with the equations developed above for lighting.

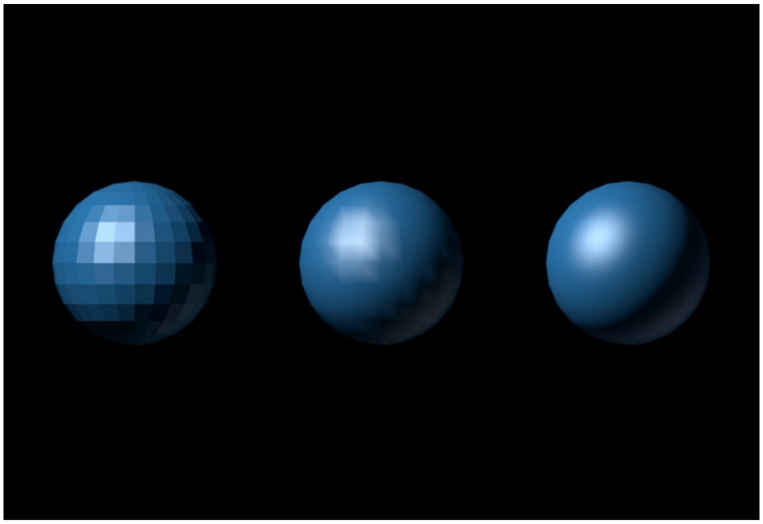

Illustration 6: A comparison of flat, Gouraud, and Phong shading techniques

Adding Color

Up until now, we considered achromatic light (shades of gray, if you prefer). Adding color is a simple process by which we consider RGB as independent channels that we each treat in the same way we did for achromatic light. For instance, the equations for the three RGB channels simply become:

$$\begin{aligned} I_R &= I_{Ra}\rho_{Ra}+I_{Rd}+I_{Rp} \\ I_G &= I_{Ga}\rho_{Ga}+I_{Gd}+I_{Gp} \\ I_B &= I_{Ba}\rho_{Ba}+I_{Bd}+I_{Bp} \end{aligned}$$

and the final intensity of the point is $I=I_R+I_G+I_B$.

Lighting Framework

In the previous sections we have examined techniques to approximate the lighting in the real word with ambient light, diffuse and specular reflection, and lastly how to combine these models into a three-channel RGB color scheme for incident light. Additionally, it is possible to define properties to materials (objects).

Material Colors

The lighting model makes the approximation that a material's color depends on the percentages of the incoming red, green, and blue light it reflects. For example, a perfectly red ball reflects all the incoming red light and absorbs all the green and blue light that strikes it. If you view such a ball in white light (composed of equal amounts of red, green, and blue light), all the red is reflected, and you see a red ball. If the ball is viewed in pure red light, it also appears to be red. If, however, the red ball is viewed in pure green light, it appears black (all the green is absorbed, and there's no incoming red, so no light is reflected).

Like light, materials have different ambient, diffuse, and specular colors, which determine the ambient, diffuse, and specular reflectances of the material. A material's ambient reflectance is combined with the ambient component of each incoming light source, the diffuse reflectance with the light's diffuse component, and similarly for the specular reflectance and component. Ambient and diffuse reflectances define the color of the material and are typically similar if not identical. Specular reflectance is usually white or gray, so that specular highlights end up being the color of the light source's specular intensity. If one thinks of a white light shining on a shiny red plastic sphere, most of the sphere appears red, but the shiny highlight is white.

In addition to ambient, diffuse, and specular colors, materials may have an emissive color, which simulates light originating from an object. The emissive color of a surface adds intensity to the object, but is unaffected by any light sources. Also, an emissive color does not introduce any additional light into the overall scene.

RGB Values for Lights and Materials

The color components specified for lights mean something different than for materials. For a light, the numbers correspond to a percentage of full intensity for each color. If the RGB values for the color of a light are all 1.0, then the light is the brightest possible white. If the values are 0.5, the color is still white, but only at half intensity, so it appears gray, and so on.

For materials, the numbers correspond to the reflected proportions of those colors. So if $R=1$, $G=0.5$, and $B=0$ for a material, then it reflects all the incoming red light, half the incoming green, and none of the incoming blue light. In other words, if a light has components (R_L, G_L, B_L) , and a material has corresponding components (R_M, G_M, B_M) then the light that reaches the eye is given by $(R_L R_M, G_L G_M, B_L B_M)$.

Similarly, if two lights send (R_1, G_1, B_1) and (R_2, G_2, B_2) to the eye, then we add the components together as $(R_1+R_2, G_1+G_2, B_1+B_2)$ and, any of the sums are greater than 1 (corresponding to a color brighter than the equipment can display), then the component is scaled back to 1.

Rendering a Sphere

These are the steps required to add lighting to a scene:

- Define normal vectors for each vertex of all the objects. These normals determine the orientation of the object relative to the light sources.
- Create, select, and position one or more light sources.
- Create and select a lighting model which defines the level of global ambient light and the effective location of the viewpoint.
- Define material properties for the objects in the scene.

Normal Vectors for Each Vertex

The face normals of an object determine the orientation of the faces relative to the light source. For each vertex, we use the assigned normal to determine how much light that particular vertex receives from each light source.

Using Light Sources

Light sources can be located wherever they are desired. They can be positioned near the scene, or at infinity such as the sun.

Lighting Models

The lighting model defines whether the viewer of the scene should be considered to be an infinite distance away or local to the scene, and whether lighting calculations should be performed differently for the front and back surfaces of objects in the scene. Since the back faces of the sphere are hidden, a front-face lighting is adequate.

Material Properties

The material properties of an object define how it reflects light and the type of material it appears to be made of. It is possible to specify the ambient, diffuse, and specular colors of an object, in addition to its shininess.

Controlling the Light Position and Direction

We treat the position and direction of a light source just as we treat the position of a geometric primitive. In other words, a light source is subject to the same matrix transformations. This means we can manipulate a light source position or direction by changing the contents of the matrix pipeline. We can achieve the following three different effects by changing the point in the program at which the light position is set, relative to viewing transformations:

- A light position that remains fixed
- A light that moves around a stationary object
- A light that moves along with the viewpoint