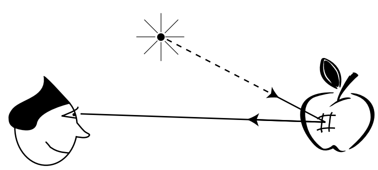
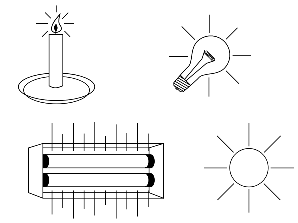
Chapter 8

Lighting and Shading

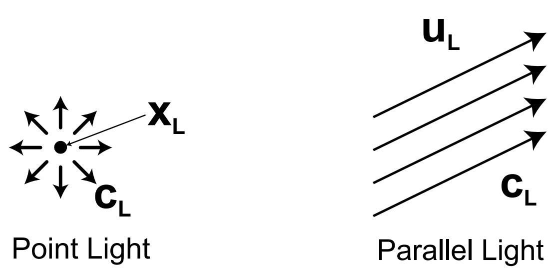


When we think that we "see" an object in the real world, we are not actually seeing the object itself. Instead, we are seeing light either emitted from or reflected off of that object. The pattern of the light reaching our eyes allows us to reconstruct, in our minds, the form of the object that we are looking at. Consequently, in order to simulate this image formation process in a computer, we need to be able to model *lights*, *materials* and *geometry*. Lights provide a source of illumination, materials determine how light is reflected off of a surface made of this material, and geometry provides the shape of the surface.

8.1 Point and parallel lights

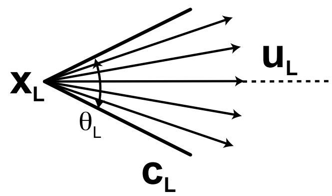


In the real world, all lights have a finite size and shape, and a location in space. However, we will see later that modeling the surface area of the light in a lighting algorithm can be difficult. It turns out to be much easier, and generally quite effective, to model lights as either being geometric points or as being infinitely far away. A point light has a single location in 3D space, and radiates light equally in all directions. An *infinite* light is considered to be so far away that all of its light rays are parallel to each other. Thus, an infinite light has no position, but there is a fixed direction for all of its rays. Such a light is sometimes called a *parallel* light source.



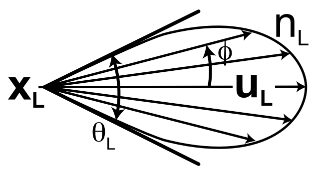
A point light is specified by its position $\mathbf{x}_L$, and a parallel light is specified by its light direction vector $\mathbf{u}_L$. All lights need a specification of the light luminance and chromatic content (i.e. its hue and color purity). We call the combination of luminance and chroma the light color, $\mathbf{c}_L$. For most applications, it is sufficient for the light color to be represented in RGB coordinates. The luminance of the light is approximated by the sum of the light's R, G, and B components (we will learn how to determine the perceptual luminance when we study color). The chroma of the light is given by the ratios of R, G, and B to the luminance. In advanced lighting systems, however, we may want to use a system that samples the spectrum of visible light more finely in order to achieve more physically accurate, subtle effects.

8.2 Spotlights



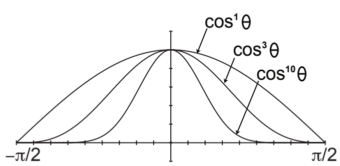
Simple Spotlight

There are two main variants of the point light. One is to provide a direction and cone of projection to model a spotlight, which is specified by providing a location $\mathbf{x}_L$, an aim direction $\mathbf{u}_L$, and the angle of a conic shade θ_L. Such a light acts just like a point light for all directions within the cone of projection, but provides no light outside of that cone.



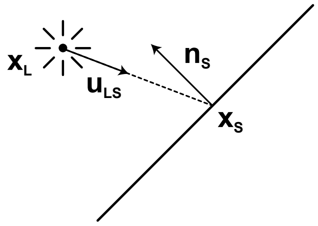
Spotlight

A spotlight can me made more realistic by providing "fall off" of light intensity with angle away from the aim direction $\mathbf{u}_L$. At angles outside of the cone, no light is visible, but from angles inside the cone the light intensity is a function of angle ϕ between the aim direction and the direction towards the point being illuminated.



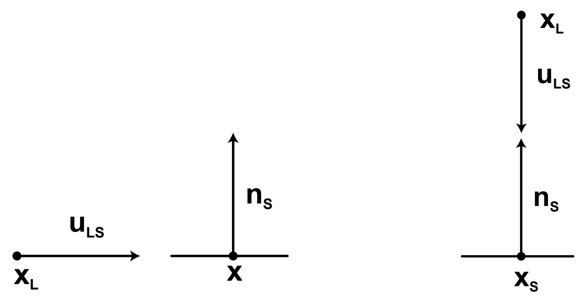
Typically, the intensity modulation function used is $\cos^{n_L}\phi$, where n_L is an exponent determining the sharpness of fall off with angle ϕ. The graph to the left shows the effect of providing an exponent. If $n_L = 1$, we get a gradual cosine-like falloff. As n_L becomes larger, the fall off is steeper, and the effect is that the spotlight becomes more focused on its aim direction.

8.3 Incident light



Now, let us compute the direction and the intensity of the *incident light* reaching a point $\mathbf{x}_s$ on a surface from a light L. For a point light, the direction is $\mathbf{u}_{Ls} = (\mathbf{x}_s - \mathbf{x}_L)/\|\mathbf{x}_s - \mathbf{x}_L\|$. The amount of light arriving at the surface around $\mathbf{x}_s$ will be proportional to the cosine of the angle between $\mathbf{u}_{Ls}$ and the surface normal $\mathbf{n}_s$ at $\mathbf{x}_s$. Thus, the incident light color will be $\mathbf{c} = -(\mathbf{u}_{Ls} \cdot \mathbf{n}_s)\mathbf{c}_L$.

To see why the incident light should be modulated by the cosine of the angle between the incident light and the surface normal, consider the two extremes:



In the left-hand figure, the light is parallel to the surface, and no light will be reflected. In the right-hand figure, the light is perpendicular to the surface and shines on the surface with full intensity. Angles in between these two extremes will result in a light intensity proportional to the cosine of the angle between light and surface normal directions.

For a parallel light, the light direction is always constant, so $\mathbf{u}_{Ls} = \mathbf{u}_L$. The color calculation will be the same as above. For a spotlight, the calculation is the same as for a point light, except $\mathbf{c} = -(\mathbf{u}_{Ls} \cdot \mathbf{u}_l)^{n_L}(\mathbf{u}_{Ls} \cdot \mathbf{n}_s)\mathbf{c}_L$ to account for the falloff of the spot strength with ϕ.

Note, in the above calculations, $\mathbf{u}_{Ls} \cdot \mathbf{n}_s = -\cos\theta$ where θ is the angle between the light direction and the surface normal. Whereas $\mathbf{u}_{Ls} \cdot \mathbf{u}_L = \cos\phi$, the angle between the light direction and the light aim vector.

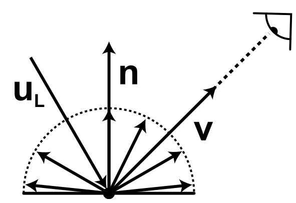
Note also, for a spotlight $\mathbf{c} = \mathbf{0}$ if $|\phi| > \theta_L/2$ (i.e, the light direction is outside of the spot's cone), and that, irrespective of the light type, if $\theta > \pi/2$ or $\theta < -\pi/2$ then $\mathbf{c} = 0$, since all of the light is shining on the *back* of the surface.

8.4 Reflected light

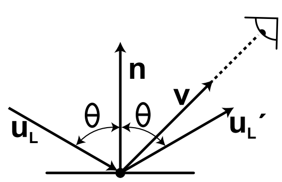
At this point, we know how to compute the light illuminating a point on a surface. Now we need to compute the color of the light reaching the eye. This will depend on several things:

1. the color illuminating the point on the surface,
2. the color reflectance properties of the material making up the surface,
3. the position of the viewer relative to the orientation of the surface.

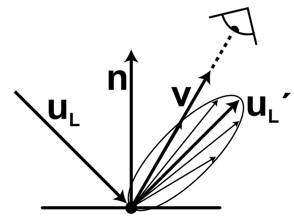
In the most general case, light hits a surface and can be scattered off of that surface in many directions. In graphics, the most common model of this is to consider two kinds of surface reflection: *diffuse reflection* and *specular reflection*. The net reflection will be a linear sum of these.



Diffuse reflection, known formally as *Lambertian reflection*, models light reflection from a matte (completely non-shiny) surface. In a Lambertian reflection, light is scattered off the surface equally in all directions, as indicated in the figure to the right. Thus, as long as the viewer, looking in the view direction $-\mathbf{v}$ can see the surface, the amount of reflected light is independent of view direction and is proportional to the light illuminating the surface. In diffuse reflection, when the incident light is scattered off of the surface, there is the assumption that the light is interacting with the surface material and picking up some of its color. Thus the color of the reflected light is modulated by the surface color. In a typical computer graphics model, the incident light is modeled as an RGB color, and the material of the surface is also modeled as an RGB color. To compute the diffusely reflected color, we simply multiply the like color components of the light and material colors (with the assumption that the colors are stored as floating point numbers between a minimum value of 0.0 and a maximum value of 1.0).



In specular reflection, it is assumed that the surface is a mirror reflector, reflecting light most strongly in the mirror reflection direction, as in the figure to the left. For a pure specular reflector, like a mirror, reflected light will only be seen when the view direction $-\mathbf{v}$ and the reflected light direction $\mathbf{u}'_L$ are exactly parallel. In specular reflection, the assumption is that the light is reflected off of a shiny layer over the actual material of the surface, so that the light does not pick up any of the material's color. Thus, the color of the reflected light is simply the light's color. Alternately, the shiny layer may have its own color, which is different from the material's diffuse reflecting color. In this case, the color of the reflected light would be modulated by the shiny layer's color.



If we assume that a surface is glossy but not a perfect mirror, we might expect some light scattering but mainly about the mirror reflection direction. This is illustrated in the figure to the right. Note, that this looks very much like how lighting is modeled with a spotlight, and can be modeled by modulating reflected light intensity by $cos^n\phi$, where ϕ is the angle between the reflection vector $\mathbf{u}'_L$ and the view vector $\mathbf{v}$. In this case, the amount of reflected light would be proportional to $(\mathbf{u}'_L \cdot \mathbf{v})^n$. The exponent n is called the *specular exponent* of the surface, and, just like with the spotlight calculation, determines the breadth of the spread of the specular reflection. Very high values of n, for example $n = 100$, produce small highly focused *specular highlights* on the surface when the viewer is exactly in the mirror reflection direction from the light source. Low values of n, for example $n = 5$, produce very broad diffuse specular highlights.

Real materials reflect light in complex ways; no material is a pure diffuse reflector or a pure specular reflector. Real surfaces can often be approximated by considering that some fraction k_d of the light will be reflected diffusely and some fraction k_s reflected specularly. With this model, the total reflected illumination would be given by

$$I = k_d I_d + k_s I_s,$$

where,

$$I_d = -(\mathbf{u}_L \cdot \mathbf{n})\mathbf{c}_L\mathbf{c}_m, \;\; I_s = (\mathbf{u}'_L \cdot \mathbf{v})^n \mathbf{c}_L\mathbf{c}_s,$$

$\mathbf{c}_L$ is the color of the light, $\mathbf{c}_m$ is the diffuse reflectance color of the material, and $\mathbf{c}_s$ is the specular reflectance color of the material. The product $\mathbf{c}_m\mathbf{c}_L$ (and $\mathbf{c}_s\mathbf{c}_L$) is done componentwise, yielding $\begin{bmatrix} \mathbf{c}_{LR}\mathbf{c}_{mR} \\ \mathbf{c}_{LG}\mathbf{c}_{mG} \\ \mathbf{c}_{LB}\mathbf{c}_{mB} \end{bmatrix}$, so these are really three equations giving the red, green, and blue components of reflected light. Usually $\mathbf{c}_s$ is taken to be white, in which case the multiply by $\mathbf{c}_s$ to compute the specular color can be dropped.

Finally, to get a complete illumination model, we must consider the light in a scene that is coming from interreflections between objects, from skylight, or from other sources of illumination not accounted for in the model. We call the total of all this light *ambient* illumination and assume that it is constant everywhere in the scene, and coming equally from all directions. We represent this by I_a in our shading equation, giving

$$I = k_a I_a + k_d I_d + k_s I_s,$$

with

$$I_a = \mathbf{c}_a,$$

which is the color of the ambient light and a constant for the scene. The constants k_a, k_d, and k_s are redundant, because their effects can be included with the colors of the lights and materials, and so they are usually all set to 1. However, they are handy to have as an independent way to adjust the ambient, diffuse, and specular components of the illumination.