# Chapter 11

# Lighting and Surfaces

## 11.1 Introduction to Lighting

Three-dimensional surfaces can be drawn as meshes or wireframe diagrams, and these are often useful. However, a wireframe turns off most of the visual cues that we use to perceive three-dimensional objects in the real world. Coloring, shading and highlighting the surface can help. To understand the possibilities, we need to understand how real objects react to light, and how computer graphics simulates this.

There are three species of light (or "illumination models"):

1. Intrinsic (self-emitting)
2. Ambient light (sometimes called "diffuse light")
3. Light sources (that is, *concentrated*, lamp-like sources)

There are two kinds of reflection:

1. Specular ("mirror-like") reflection
2. Diffuse ("Lambertian") reflection

When the surface is approximated as the union of a lot of flat-sided or non-planar polygons, there are three kinds of shading:

1. Flat (also called, more precisely, "faceted")
2. Gourard
3. Phong

There are two strategies for computing the total light environment:

1. Ray-tracing
2. Radiosity

There are several strategies for applying a texture to a surface including:

1. Texture mapping
2. Bump mapping
3. Reflection mapping

In this chapter, we will discuss each of these in turn.

## 11.2 Lighting or Illumination Models

A graphics window may display many objects, some of which are brightly illuminated and others which are in the shadows. It follows that to discuss lighting models, we must take it one-object-at-a-time.

“Intrinsic” light is the light emitted by the object itself, such as the glow from a TV screen, a light-emitting diode, or a star.

“Ambient” light is an illumination that seems to come from all sides. In the real world, light scatters off the walls of a room (or the ground) and then bounces off an object and then to our eyes. In scientific visualization, ambient light is usually assumed to be uniform, that is, to be coming with equal intensity from all directions. In photorealistic graphics, the ambient light is allowed to vary so that it is relatively intense in the directions subtended by walls painted in glossy white and relatively weak from dark, light-absorbing wood paneling. Ambinet light is sometimes also called “diffuse” light, which is rather misleading because it has nothing to do with diffusion.

“Light sources” are lamps, but lamps in the narrow sense of a source of once-scattered light that bounces directly off the object into our eyes. A real-world lamp is usually a source of both once-scattered light as well as multiply-scattered light. By convention, “light source” is used to describe the once-scattered light only while “ambient light” is applied to light that bounces off something other than the graphic object before reflecting, in the final scattering, from the object itself.

In photorealistic graphics, light sources can be extended objects, that is, the graphics calculation must allow for the fact that a fluorescent overhead light may be several feet long, and is most effective at lighting objects directly underneath. In scientific graphics, it is usual to pretend that all “light sources” are *point* sources, sufficiently far from the object that all rays fall onto it from a single angle in a spherical coordinate system centered on the middle of the object.

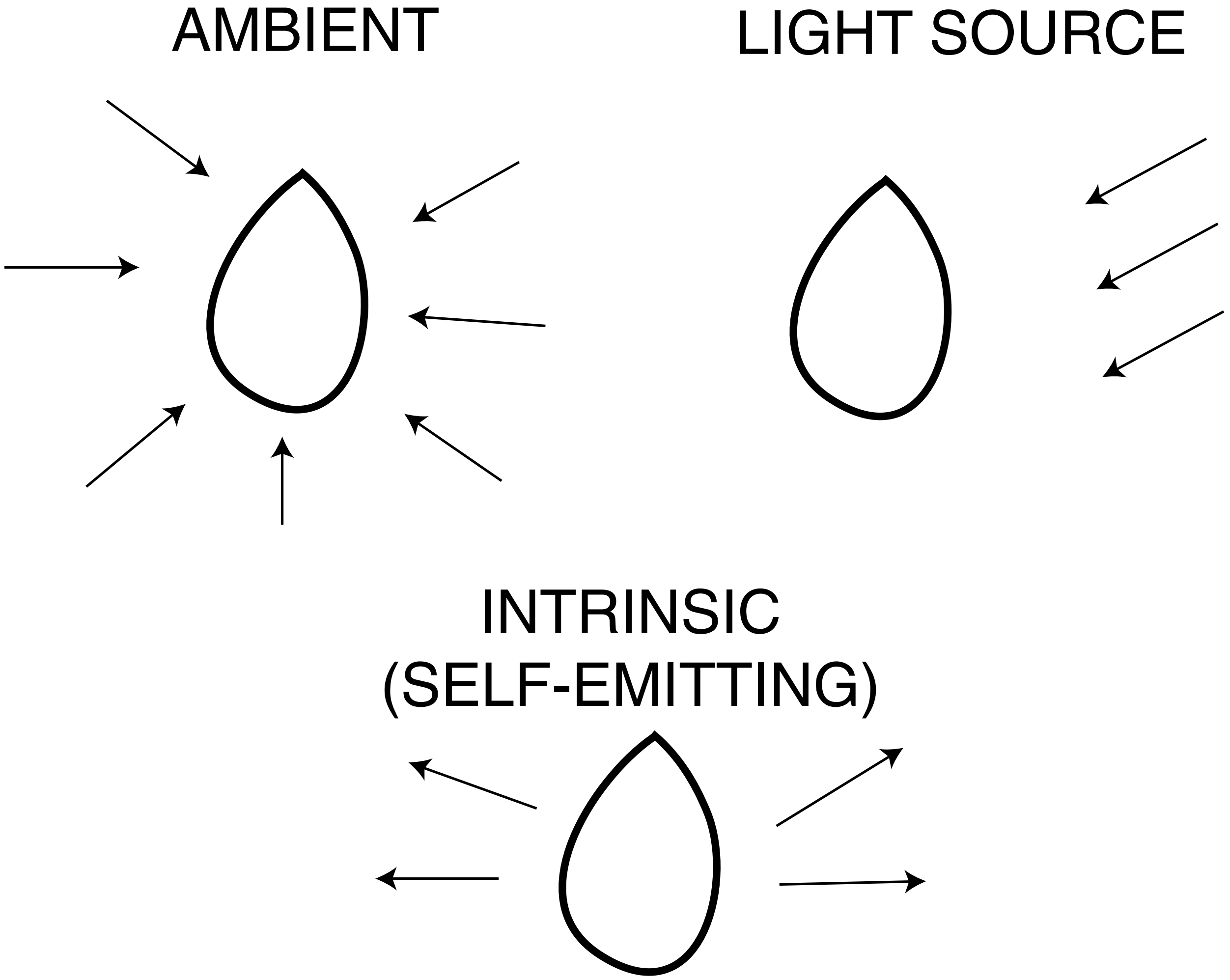

Figure 11.1: Schematic of the three sources of lighting for an object.

## 11.3 Reflection

The scattering of light from an object is usually described by considering two extreme cases: specular reflection and diffuse reflection. Real objects reflect light as a mixture of these two extremes, but by adding a little bit of diffuse reflection to a little bit of specular reflection in some object-dependent fashion, one can obtain a good approximation to the scattering from almost anything.

"Specular" is derived from the Latin "speculum", which means "mirror". If a line is drawn perpendicular to the surface (the "surface normal", shown in Fig. 11.2 as a dashed line) and the incoming ray makes an angle of $\theta$ with the normal, then all the reflected light will also make an angle of $\theta$ with the normal but on the opposite side.

If the object has a curved surface, then the eye will see only that part of the surface where the angle condition is satisfied. The bright reflection from that small part of the surface is a "highlight".

Real objects are never perfectly mirror-like; under a microscope, every surface is pitted with small peaks and valleys. These bumps scatter the light at all angles as illustrated in Fig. **??**. The intensity of the scattered light is

$$I_{Lambert} = I_{source}\, k_d \, \cos(\theta) = I_{source}\, k_d \, (\mathbf{N} \cdot \mathbf{L}) \tag{11.1}$$

where $I_{source}$ is the intensity of light from the source (as measured by the energy falling on a plane perpendicular to the incoming rays), $k_d$ is the "diffuse-reflection" or "Lambertian coefficient", $\theta$ is the angle between the surface normal and the incoming light from the source; in the alternative form, **N** and **L** are the unit normal and a unit vector pointing at the light source, respectively, with the dot denoting the usual inner product of two vectors.

Lambertian reflection gives a "matte" finish to an object.

It is common to fudge the appearance of diffuse reflection by the empirical artifice of adding an exponent to the $\cos(\theta)$ term to modify the calculated reflection to

$$I_{Lambert} = I_{source}\, k_d \, \{\cos(\theta)\}^{n_d} \tag{11.2}$$

where $n_d$ is the "diffuse reflection exponent" or "Lambertian exponent". Taking $n_d < 1$ gives a broader, more diffuse ring of light from the object whereas $n_d > 1$ gives a more concentrated ring of light than the default $n_d = 1$.

A major difference between the two types of reflection is that in their purest form, the color of Lambertian reflection is the *color* of the *object*. In contrast, in specular reflection, the default color of the reflection is the *color* of the *light source*.

The specular-is-source-color assumption is accurate for plastic, but inaccurate for metal and velvet. The highbrow way to correct for this is to use either the Torrance-Sparrow or Cook-Torrance lighting models, which use a theory of microfacets and some physics. A lowbrow way is to provide the user with a tweakable parameter that makes the reflected color a weighted average of the object color and the light source color.

The usual (simplest) formula for specular reflection is

$$I_{specular} = I_{source}\, k_s \, (\mathbf{N} \cdot \mathbf{H})^{n_s} \tag{11.3}$$

where $k_s$ is the specular reflection coefficient (which depends on the material the object is made of), $n_s$ is the specular reflection exponent, and **H** is the "halfway" vector which is halfway between **L**, the vector pointing at the light source, and a vector **V** that points at the viewer's eye. If **H** is parallel to the surface normal **N**, then the viewer will see the highlight, so **H** is the known as the vector of "maximum highlights". (For a perfect mirror, **H** would be more precisely the vector of "only highlights" since that point of the surface would be invisible if **H** did not parallel **N**, but real lights are not point lights, and real surfaces are not perfectly mirror-like.)

## 11.4 Shading

A scheme for applying color to the polygons of a surface is called a "shading" model. It might equally well be called a "coloring" model except that the methods are equally applicable to grayscale.

The simplest model is called "flat" or "faceted" shading. Flat shading means that the color or grayscale is uniform within each polygon. This is computationally cheaper than any of the other alternatives (except setting the face color equal to none, and thus reducing the surface to a mesh). However, unless the polygons are very, very numerous, there will be obvious discontinuities of color across the edges of each polygon.

In photorealistic graphics, this is almost always a BAD THING. In scientific or engineering graphics, it may be okay. If the object is a crystal, for example, the flat-sided facets are real, and color discontinuities actually make it easier for the viewer to perceive the true shape of the crystal.

Similarly, scientific graphics almost always allow the user the option of a black mesh to emphasize the edges between different polygons. Even when the model object has a smooth surface, the mesh and flat shading may be useful in identifying the subdomains of the finite element or spectral element that was used to compute fluid flow or elasticity or other properties of the surface.

Gouraud shading is the next step up in both photorealism and computational cost from flat shading. (This method is named after its inventor, Henri Gouraud, whose name rhymes with "Thoreau".) The first step is to compute the vertex normals, which are the normals to the surface at the vertices. Of course, a vertex is a "point", which casts serious aspersions on the concept of "perpendicular", which can be defined only relative to a surface. The vertex normal is defined by computing the normals to the surfaces of each of the faces that meet at a given vertex and then taking the average.

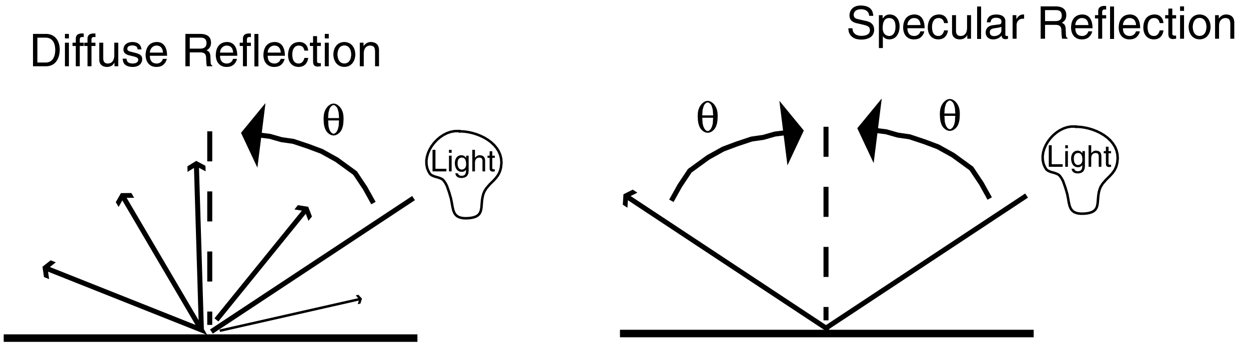

Figure 11.2: Two species of reflection. In diffuse or Lambertian reflection, the light scatters in all directions. However, the intensity of the reflection light is proportional to the intensity of the source multiplied by $\cos(\theta)$, multiplied by a constant $k_d < 1$ that depends on the reflectivity of the object. $\theta$ is the angle between the incoming rays from the source and the surface normal, which is shown as the dashed line perpendicular to the surface, which is the horizontal thick line. In specular or "mirror-like" reflection, the reflected light is concentrated in a narrow angle which makes the same angle $\theta$ with the surface normal.

The vertex normal provides the angle $\theta$ between the surface and the incoming rays from the light source. The second step in Gouraud shading is to apply the formulas given in the previous section to compute the Lambertian and specular reflection for each vertex. The third step is to interpolate the reflection between the reflections at each of the vertices that define a given face so as to compute a smoothly varying reflection for each pixel on a given polygon. The color discontinuities are completely eliminated.

Phong shading, originated by Phon Bui-tuong, is much more expensive than Gouraud shading. The reflectance is calculated at many sample points on a given face instead of just at the vertices. The result is a more realistic shading. However, the improvement is often small, so Gouraud shading is still widely used.

Scientific and engineering visualization, because they rarely strive for photorealism, does not desperately need the extra fidelity of Phong shading. However, because the rendering times for scientific graphics are rather short, again because of the lack of photorealism, the price for Phong shading is usually only an extra second or two.

## 11.5 Ray-tracing and Radiosity

Ray-tracing and radiosity are both schemes of "light bookkeeping". In the first method, rays are traced from the light source to various objects and thence by reflection to the viewer's eye, or in reverse from the eye to the light source. (Reverse, eye-to-source ray-tracing is the more common than direct tracing because it omits rays from the source that are reflected away from the viewer and thus are irrelevant to the displayed image.)

Radiosity eschews rays in favor of breaking down all objects (including the walls and other contributors to the ambient light) into lots and lots of patches. Each patch is then treated as an emitter or a reflector. The radiosity method then computes the interactions of each patch with every other patch. Radiosity software is rather complicated because it employs a lot of tricks to ignore the interactions of distant patches with one another, and thereby reduce rather humongous calculations to a managable length.

Ray-tracing is computationally intensive, too. One major difference between the two methods is that ray-tracing is *view-dependent*. If the viewing angle is *changed*, then all ray-tracing calculations must be redone from scratch. In contrast, radiosity is *view-independent* in the sense that only a small fraction of the overall calculations must be repeated when the viewing angle is changed. Radiosity is very popular in virtual reality simulations and in architecture computed-aided design. For the latter, for example, the computer can precalculate most of the numbers it needs to present an entire building. The client can then take a virtual walk-through the building; the workstation is able to present changes of view so quickly that the client does not become bored and distracted between scenes because the computer only has to recalculate a tiny portion of the overall radiosity computations each time the view is changed.

The other major difference is that ray-tracing is oriented towards *specular* reflection; ambient light can always be modelled as the union of $N$ light sources where $N$ is sufficiently large, but the work of ray-tracing ambient light is then $N$ times the labor of tracing a single light source. The radiosity method, in contrast, is optimized for diffuse reflection and ambient light. The light in most architectural structures tends to be rather diffuse, which is another reason that architectural graphics are dominated by the radiosity method.

In an environment with both light sources and ambient light, ray-tracing and radiosity can be combined. Some applications use radiosity for non-reflective surfaces and ray-tracing for reflective surfaces.