

# Light and Shading

An *illumination model* (also *lighting model* or *shading model*) is used for determining the color and brightness a viewer perceives and therefore which color a pixel should have, according to the lighting conditions and surface-properties of the objects. Together with the perspective projection, this is one of the most important contributions to the realistic look of computer-generated images.

To keep things simple, all of the following examinations and formulas use only light intensity. To handle colors, all the calculations have to be done in several wavelengths. The simplest case is to do this with red, green and blue.

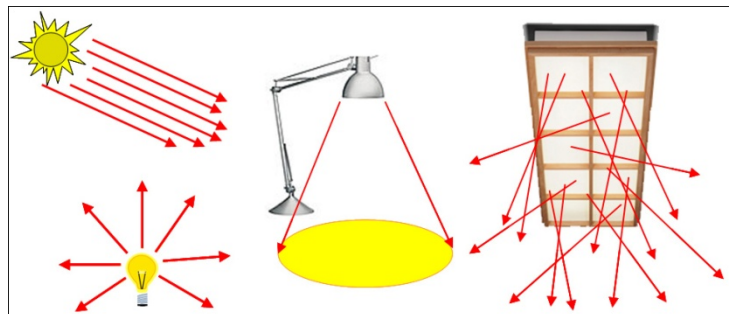
## ■ Light Sources and Surfaces

### **Light Source**

To calculate the influence of light, light sources are needed. Properties of light sources can be:

*Form:* directional light, point light source, directional point light, area light source, ...

*Properties:* brightness, color, distance, ...



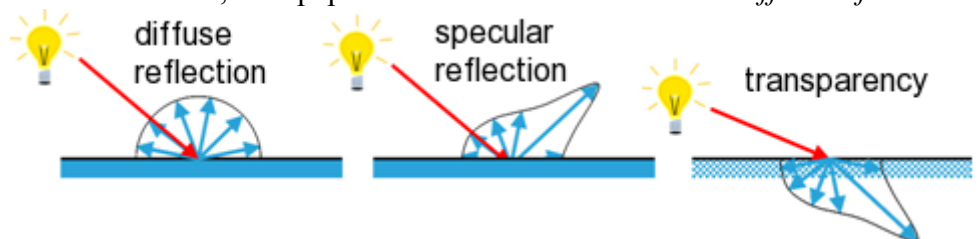
### **Object surfaces**

Surfaces can reflect light equally in all directions, like paper or chalk do. This is called *diffuse reflection*.

Surfaces can also reflect most of the light in a particular direction, like metal, varnish or polish. This is called *specular reflection*.

Furthermore, surfaces can be *transparent*, which means that light goes

through the surface and leaves it on the other side, as with glass or water. Real surfaces usually have a mixture of these properties. Note that light not only incidents from light sources, but also from reflections from other surfaces.



## ■ A Simple Lighting Model

Physically exact simulation of light and its interaction with surfaces is very complex. Therefore, in practice, simplified and empirical lighting models are used, which are constructed as follows.

### **Background Light (Ambient Light)**

Since every object reflects a part of the incoming light, no object is completely dark in areas where there is no direct light incident from a light source. This everywhere existent light is called background light, or *ambient light*. In simple illumination models a constant value  $I_a$  is included in the lighting calculations for the ambient light.



### Lambert's Law

This law states, that the flatter light incidents on a surface, the darker this surface appears. Through this effect we get the impression of a spatial form.

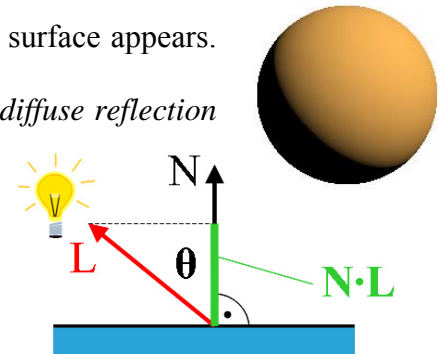
Let  $I_1$  be the brightness of the involved light source and  $k_d$ ,  $0 \leq k_d \leq 1$ , the *diffuse reflection coefficient* which indicates what percentage of incoming light is reflected equally in all directions. Furthermore let  $\theta$  be the angle between the surface normal and the direction to the light source, which is the direction of incident light. Then the resulting intensity  $I$  at the surface-point is:

$$I = k_d \cdot I_1 \cdot \cos \theta = k_d \cdot I_1 \cdot N \cdot L$$

[ $N \cdot L$  means scalar product]

If in addition ambient light is added, a nice sphere is the result

(upper sphere = only diffuse light, lower sphere = diffuse + ambient).

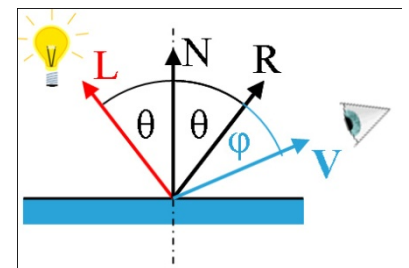
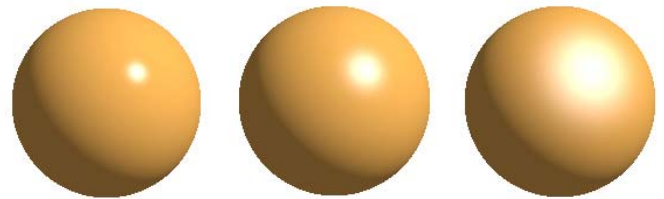


### Specular Highlights

Almost all surfaces are slightly reflecting. If this is not modeled in an illumination model, the objects appear dull. Because the exact computation of reflections is quite complicated to calculate, a simple function, which has similar characteristics as the highlight, is used as an approximation instead. The function used is  $\cos^n$ . With the free parameter  $n$  the glossiness of the surface can be controlled. The larger  $n$ , the smaller the highlight and therefore the surface appears "shiny" or "more polished" (left sphere). The smaller  $n$  is, the duller the surface appears (right sphere). To add the highlights to the lighting model, the *specular reflection coefficient*  $k_s$  is introduced. According to the *Phong-illumination-model* the highlight is then calculated as follows:

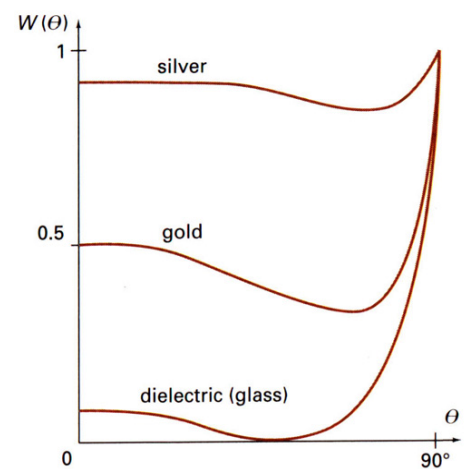
$$I_{i,spec} = k_s \cdot I_1 \cdot \cos^n \varphi = k_s \cdot I_1 \cdot (R \cdot V)^n$$

The angle  $\varphi$  is the difference between the exact reflection ray and the direction to the eye.



A more physically correct model is obtained by using the Fresnel equations for reflectance, which describe the dependency of reflectance on the angle of light incidence. This means that the coefficient  $k_s$  is actually a function  $W(\theta)$  of the light incidence direction. For most materials however, this value is almost constant. This is the reason why usually this more complex model is not applied, unless a material is used where this behavior is noticeable. The image on the right shows the dependence of the function  $W(\theta)$  on the angle between incident light and the surface normal for three different materials.

When calculating the vector  $R$  it should be noted, that all vectors are in 3D space.  $L$ ,  $N$  and  $R$  must lie in one plane and all of them have to have unit length (1).  $R$  can be calculated as  $R = (2N \cdot L) N - L$ . Since the function for the highlights is just a rough estimation, often a simpler formula is used, where  $R \cdot V$  is replaced by  $N \cdot H$ . The angle between  $N$  and the bisector  $H$  between  $L$  and  $V$  is almost proportional to  $\varphi$ .



Combining all those lighting components, we get

There are many more aspects which have to be considered to get closer to “real” images, but this will not be described here: color-shift depending on the view-direction, influence of the distance to the light source, anisotropic surfaces and light sources, transparency, atmospheric effects, shadows, etc.

## ■ Shading of Polygons

### Flat-Shading

When shading a polygon, clearly every point on it has the same surface-properties, in particular the same normal vector. If every polygon is simply filled with one color, the edges between the polygons become even more visible and appear unaesthetic (and they are wrong!). The mechanism of the eye which amplifies edges enhances the appearance of these unwanted edges, it is known as the Mach-Band effect. This effect lets us perceive the dark side of each polygon even darker and the bright side even brighter at the edges than they really are. The simplest solution to this is the interpolation of the shading between the polygons. To do this, there are two common approaches: Gouraud-shading and Phong-shading.

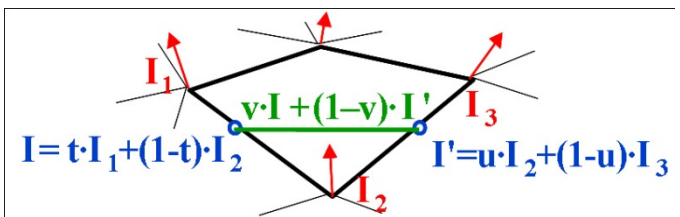


### Gouraud-Shading

When applying Gouraud-shading, the calculated brightness values are interpolated over the polygon areas. For this purpose, brightness values are calculated at the vertices of the polygon and then they are linearly interpolated over the polygon. More exactly it is done as follows:

(1) At each vertex, a normal is calculated as the average of the normals of the adjacent polygons. This is an estimation of the normal of the underlying surface.

(2) From the properties of the surface, the normal and the direction of incident light, a brightness-value (“shading”) is calculated. Note that all polygons adjacent to a vertex have the same values at this vertex.



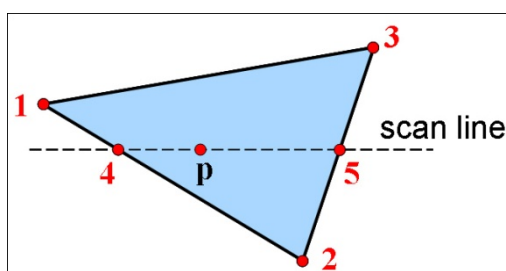
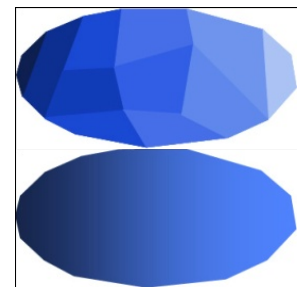
(3) Along the edges of a polygon the brightness-values are linearly interpolated, that is, for every intersection point with a scan-line the values are determined. Note that this results in equal values along the common edge of neighboring polygons.

(4) Along each scanline the values are again linearly interpolated from the left to the right polygon boundary. In this way the values of neighboring pixels are very similar and no edges are visible.



Nevertheless some errors remain. One of them is that the silhouette does not change and there are still annoying polygon-edges at the boundaries (see image on the right). Furthermore, more or less arbitrary results are achieved in the area of highlights, depending on whether any normal happens to create a highlight or not.

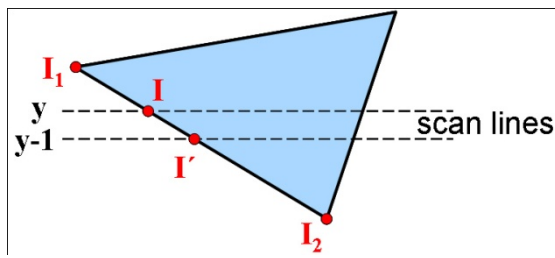
This is especially disturbing for moving objects.



$$I_4 = \frac{y_4 - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y_4}{y_1 - y_2} I_2$$

$$I_p = \frac{x_5 - x_p}{x_5 - x_4} I_4 + \frac{x_p - x_4}{x_5 - x_4} I_5$$

The (linear!) *interpolation of intensities* can again be done incrementally:

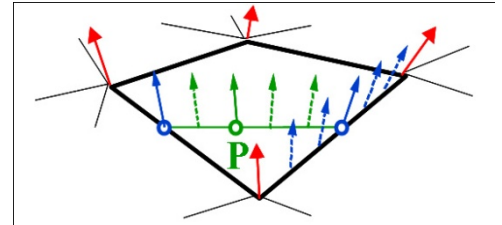


$$I = \frac{y - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y}{y_1 - y_2} I_2$$

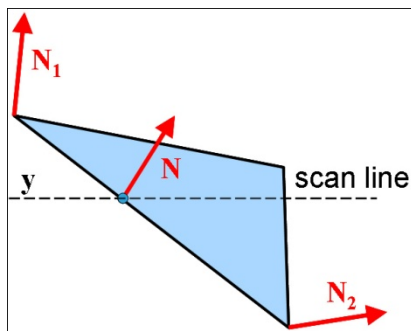
$$I' = I + \frac{I_2 - I_1}{y_1 - y_2}$$

### Phong-Shading

As an alternative to Gouraud-shading, Phong-shading (not to be confused with the Phong illumination model!) creates much more consistent highlights. Similar to Gouraud-shading, interpolated normals are calculated at the vertices of each polygon. But now the normals (no brightness values calculated yet!) are interpolated along the polygon edges, and then along the scan-lines. Finally, the brightness is calculated for each single pixel according to some shading model. This approach is more complex, but leads to better results.



*Normal-vector interpolation:*



$$N = N_1(y - y_2)/(y_1 - y_2) + N_2(y_1 - y)/(y_1 - y_2)$$

### KEEP IN MIND:

The Phong illumination model (also Phong shading model) and Phong-shading (or Phong-interpolation) are two completely different things!