

# CS380 Introduction to Computer Graphics

## Homework #5

20160042 Inyong Koo

---

Describe the following terms with respect to computer graphics.

Interactions between light and materials can be classified into three groups: specular, diffuse, and translucent surface.

### 1. Specular surface

**Specular surfaces** appear shiny because most of the light that is reflected or scattered is in a narrow range of angles close to the angle of reflection. Mirrors are perfectly specular surfaces; the light an incoming light ray may be partially absorbed, but all reflected light from a given angle emerges at a single angle, obeying the rule that the angle of incidence is equal to the angle of reflection.

### 2. Diffuse surface

**Diffuse surfaces** are characterized by reflected light being scattered in all directions. Walls painted with matte or flat paint are diffuse reflectors, as are many natural materials. Perfectly diffuse surfaces scatter light equally in all directions, and thus a flat perfectly diffuse surface appears the same to all viewers.

### 3. Translucent surface

**Translucent surfaces** allow some light to penetrate the surface and to emerge from another location on the object. This process of refraction characterizes glass and water. Some incident light may also be reflected at the surface.

### 4. Ambient light

We can postulate an ambient, uniform intensity at each point in the environment. This uniform lighting is called **ambient light**.

### 5. Penumbra

A source of finite size creates shadows. Some areas are fully in shadow, or in the **umbra**, whereas others are in partial shadow, or in the **penumbra**.

### 6. Phong reflection model

The Phong reflection model has proved to be efficient and to be a close-enough approximation to physical reality to produce good renderings under a variety of lighting conditions and material properties. The Phong model uses four vectors to calculate a color for an arbitrary point  $\mathbf{p}$  on a surface;  $\mathbf{n}$ ,  $\mathbf{v}$ ,  $\mathbf{l}$ , and  $\mathbf{r}$ . The vector  $\mathbf{n}$  is the normal at  $\mathbf{p}$ . The vector  $\mathbf{v}$  is in the direction from  $\mathbf{p}$  to the viewer or COP. The vector  $\mathbf{l}$  is in the direction of a line from  $\mathbf{p}$  to an arbitrary point on the source for a distributed light source or, as we are assuming for now, to the point-light source. Finally, the vector  $\mathbf{r}$  is in the direction that a perfectly reflected ray from  $\mathbf{l}$  would take. Note that  $\mathbf{r}$  is determined by  $\mathbf{n}$  and  $\mathbf{l}$ .

The Phong model supports the three types of material-light interactions; ambient, diffuse, and specular. If we let diffuse term  $L$  and reflection term  $R$ , we can represent total intensity  $I = I_a + I_d + I_s = L_a R_a + L_d R_d + L_s R_s$ .

## 7. A shininess coefficient

The amount of light that the viewer sees depends on the angle  $\phi$  between  $\mathbf{r}$  and  $\mathbf{v}$ . The Phong model uses the equation  $I_s = L_s k_s \cos^\alpha \phi$ . The coefficient  $k_s$  ( $0 \leq k_s \leq 1$ ) is the fraction of the incoming specular light that is reflected. The exponent  $\alpha$  is a **shininess coefficient**.

## 8. Halfway angle

The **halfway angle**  $\psi$  is the angle between  $\mathbf{n}$  and  $\mathbf{h}$  (halfway vector - the unit vector halfway between the  $\mathbf{v}$  and  $\mathbf{l}$ ). When  $\mathbf{v}$  lies in the same plane as do  $\mathbf{l}$ ,  $\mathbf{n}$ , and  $\mathbf{r}$ , we can show that  $2\psi = \phi$ .

## 9. Blinn-Phong model

**Blinn-Phong model** is the lighting model which uses the halfway vector in the calculation of the specular term. This model is the default in systems with a fixed-function pipeline.

## 10. Tangent plane

The **tangent plane** gives the local orientation of the surface at a point; we can derive it by taking the linear terms of the Taylor series expansion of the surface at  $\mathbf{p}$ . The result is that at  $\mathbf{p}$ , lines in the directions of the vectors represented by

$$\frac{\partial \mathbf{p}}{\partial u} = \begin{bmatrix} \frac{\partial x}{\partial u} \\ \frac{\partial y}{\partial u} \\ \frac{\partial z}{\partial u} \end{bmatrix}, \quad \frac{\partial \mathbf{p}}{\partial v} = \begin{bmatrix} \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial v} \end{bmatrix}$$

lie in the tangent plane. We can use their cross product to obtain the normal  $\mathbf{n} = \frac{\partial \mathbf{p}}{\partial u} \times \frac{\partial \mathbf{p}}{\partial v}$ .

## 11. Angle of reflection

An ideal mirror is characterized by the following statement: *The angle of incidence is equal to the angle of reflection*. The **angle of incidence** is the angle between the normal and the light source (assumed to be a point source) - angle between  $\mathbf{n}$  and  $\mathbf{l}$ . The **angle of reflection** is the angle between the normal and the direction in which the light is reflected - angle between  $\mathbf{n}$  and  $\mathbf{r}$ .

## 12. Flat shading

If the three vectors  $\mathbf{l}$ ,  $\mathbf{n}$ , and  $\mathbf{v}$  are constant, then the shading calculation needs to be carried out only once for each polygon, and each point on the polygon is assigned the same shade. This technique is known as **flat**, or **constant, shading**.

## 13. Gouraud shading

Consider a mesh constructed with multiple polygons. Because multiple polygons meet at interior vertices of the mesh, each of which has its own normal, the normal at the vertex is discontinuous. Although this situation might complicate the mathematics, Gouraud realized that the normal at the vertex could be *defined* in such a way to achieve smoother shading through interpolation. In **Gouraud shading**, we determine color of each vertex using their vertex normal. Once smooth shading is enabled, OpenGL interpolates the colors across the faces of the polygons automatically.

## 14. Phong shading

Even the smoothness introduced by Gouraud shading may not prevent the appearance of Mach bands. Phong proposed that instead of interpolating vertex intensities, as we do in Gouraud shading, we interpolate normals across each polygon. After we compute vertex normals by interpolating over the normals of the polygons that share the vertex, we can interpolate the normals over the polygon. In terms of the pipeline, **Phong shading** requires the lighting model be applied to each fragment, hence, the name per-fragment shading.

## 15. Mach bands

The human visual system has a remarkable sensitivity to small differences in light intensity, due to a property known as lateral inhibition. If we see an increasing sequence of intensity, we perceive the increases in brightness as overshooting on one side of an intensity step and undershooting on the other. We see stripes, known as **Mach bands**, along the edges.

## 16. Vertex normal

In Gouraud shading, we define the normal at a vertex, or **vertex normal**, to be the normalized average of the normals of the polygons that share the vertex.

## 17. Per-fragment lighting

In order to obtain a smoother shading, we can do the lighting calculation on a per-fragment basis rather than on a per-vertex basis. With a fragment shader, we can do an independent lighting calculation for each fragment. This is called **per-fragment lighting**.

## 18. Nonphotorealistic shading

Programmable shaders make it possible to not only incorporate more realistic lighting models in real time but also to create interesting nonphotorealistic effects. Two such examples are the use of only a few colors and emphasizing the edges in objects. Both these effects are techniques that we might want to use to obtain a cartoonlike effect in an image.

## 19. Global lighting model (or global illumination)

Phenomena such as shadows, reflections, blockage of light are global effects and require a **global lighting model**. By using rendering strategies, including ray tracing and radiosity, we can handle global effects.