



Last updated: Yeh-Liang Hsu (2010-12-12).

Note: This is the course material for “ME550 Geometric modeling and computer graphics,” Yuan Ze University. Part of this material is adapted from *CAD/CAM Theory and Practice*, by Ibrahim Zeid, McGraw-Hill, 1991. This material is be used strictly for teaching and learning of this course.

Visual Realism

1. Shading models

Line drawings, still the most common means of communicating the geometry of mechanical parts, are limited in their ability to portray intricate shapes. **Shaded color images convey shape information that cannot be represented in line drawings. Shaded images can also convey features other than shape such as surface finish or material type** (plastic or metallic look).

Shaded-image-rendering algorithms filter information by displaying only the visible surface. Many spatial relationships that are unresolved in simple wireframe displays become clear with shaded displays. Shaded images are easier to interpret because they resemble the real objects. Shaded images also have viewing problems not present in wireframe displays. Objects of interest may be hidden or partially obstructed from view, in which case various shaded images may be obtained from various viewing points. Critical geometry such as lines, arcs, and vertices are not explicitly shown. Well-known techniques such as shaded-image/wireframe overlay (Figure 1), transparency, and sectioning can be used to resolve these problems.

◇Assignment 1

Generate an object with curved surfaces then render it using your CAD software. Are there any special functions in your CAD software that create shaded-image/wireframe overlay, transparency, and sectioning images? ◇

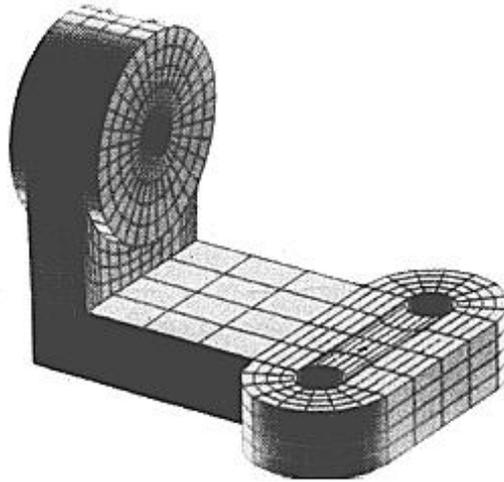


Figure 1. Shaded-image/wireframe overlay

1.1 The pinhole camera model

One of the most challenging problems in computer graphics is to generate images that appear realistic. The demand for shaded images began in the early 1970s when memory prices dropped enough to make the cost of raster technology attractive compared to the then-prevailing calligraphic displays. A three-dimensional model can be displayed by assigning different degrees of shading to the surfaces, known as **shading or rendering**. A virtual light source is assumed, and various shading techniques are available to determine strikes on each portion of the surfaces to provide a realistic image of the object.

Rendering begins by solving the hidden surface removal problem to determine which objects and/or portions of objects are visible in the scene. As the visible surfaces are found, they must be broken down into pixels and shaded correctly. **This process must take into account the position and color of the light sources and the position, orientation, and surface properties of the visible objects.**

Shading models simulate the way visible surfaces of objects reflect light. They determine the shade of a point of an object in terms of light sources, surface characteristics, and the positions and orientations of the surfaces and sources. Two types of light sources can be identified: **point light source and ambient light**. Objects illuminated with only point light source look harsh because objects are illuminated from one direction only. This produces a flashlight-like effect in a black room. **Ambient light is a light of uniform brightness and is caused by the multiple reflections of light from the many surfaces present in real environments.**

Let us examine the interaction of light with matter to gain an insight into how to develop shading models. Particularly, we consider point light sources shining on surfaces of objects. **Ambient light adds a constant intensity value to the shade at every point. The light reflected off a surface can be divided into two components: diffuse and specular. When light hits an ideal diffuse surface, it is reradiated equally in all directions**, so that the surface appears to have the same brightness from all viewing angles. **Dull surfaces exhibit diffuse reflection.** Examples of real surfaces that radiate mostly diffuse light are chalk, paper, and flat paints. **Ideal specular surfaces reradiate light in only one direction, the reflected light direction. Examples of specular surfaces are mirrors and shiny surfaces.** Physically, the difference between these two components is that diffuse light penetrates the surface of an object and is scattered internally before emerging again while specular light bounces off the surface.

The light reflected from real objects contains both diffuse and specular components, and both must be modeled to create realistic images. **A basic shading model that incorporates both a point light source and ambient light can be described as follows:**

$$\mathbf{I}_p = \mathbf{I}_d + \mathbf{I}_s + \mathbf{I}_b \quad (1)$$

where \mathbf{I}_p , \mathbf{I}_d , \mathbf{I}_s , and \mathbf{I}_b are respectively the resulting intensity at point \mathbf{P} the intensity due to the diffuse reflection component of the point light source, the intensity due to the specular reflection component, and the intensity due to ambient light.

◇Assignment 2

Put shading on the object you created in Assignment 1. Can you adjust the light source to adjust the intensities of diffuse light, specular light, and ambient light in your CAD software? Draw several pictures to show the effect of the adjustments. ◇

1.2 Diffuse reflection

Lambert's cosine law governs the diffuse reflection. It relates the amount of reflected light to the cosine of the angle θ between $\hat{\mathbf{I}}$ and $\hat{\mathbf{n}}$. Lambert's law implies that the amount of reflected light seen by the viewer is independent of the viewer's position. As shown in Figure 2, the diffuse illumination is given by

$$I_d = I_L K_d \cos \theta \quad (2)$$

where I_L and K_d are the intensity of the point light source and the diffuse-reflection coefficient respectively. K_d is a constant between 0 and 1 and varies from one material to another. Replacing $\cos \theta$ by the dot product of $\hat{\mathbf{I}}$ and $\hat{\mathbf{n}}$, we can rewrite Equation (2) as

$$I_d = I_L K_d (\hat{\mathbf{n}} \cdot \hat{\mathbf{I}}) \quad (3)$$

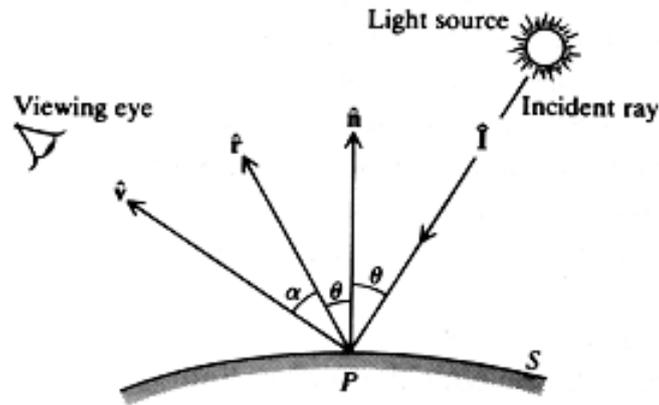


Figure 2. The geometry of shading a point.

Note that since diffuse light is radiated equally in all directions, the position of the viewing eye is not required by the computations, and **the maximum intensity occurs when the surface is perpendicular to the light source**. On the other hand, **if the angle of incidence θ exceeds 90, the surface is hidden from the light source and I_d must be set to zero**.

1.3 Specular reflection

Specular reflection is a characteristic of shiny surfaces. **Highlights visible on shiny surfaces are due to specular reflection** while other light reflected from these surfaces is caused by diffuse reflection. If you illuminate an apple with a bright light, you can observe the effects of specular reflection. Note that at the highlight the apple appears to be white (not red), which is the color of the incident light.

One of the reasonable approximations to the specular component is an empirical approximation and takes the form

$$I_s = I_L W(\theta) \cos^n \alpha \quad (4)$$

For real objects, as the angle of incidence (θ) changes, the ratio of incident light to reflected light also changes, and $W(\theta)$ is intended to model the change. In practice, however, $W(\theta)$ has been ignored by most implementors or very often is set to a constant K_s , which is selected experimentally to produce aesthetically pleasing results.

The value of n is the shininess factor and typically varies from 1 to 200, depending on the surface. As shown in Figure 3, for a perfect reflector, n would be infinite. $\text{Cos}^n \alpha$ reaches a maximum when the viewing eye is in the direction of $\hat{\mathbf{r}}(\alpha = 0)$.

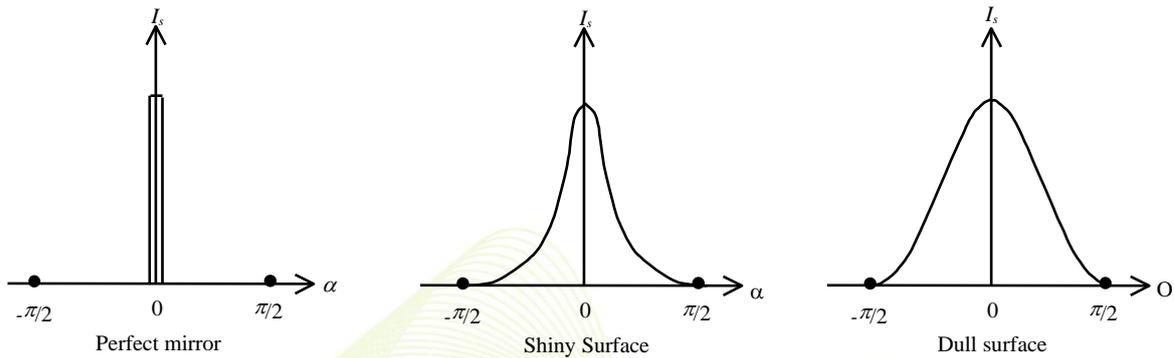


Figure 3. The reflectance of various surfaces as a function of α .

1.4 Ambient light

Ambient light is a light with uniform brightness. It therefore has a uniform or constant intensity I_a . The intensity at point \mathbf{P} due to ambient light can be written as:

$$I_b = I_a K_a \tag{5}$$

where K_a is a constant which ranges from 0 to 1. It indicates how much of the ambient light is reflected from the surface to which point \mathbf{P} belongs. Finally, considering both specular reflection and ambient light, the intensity at point \mathbf{P} .

$$I_p = I_a K_a + I_L [K_d (\hat{\mathbf{n}} \cdot \hat{\mathbf{I}}) + W(\theta) (\hat{\mathbf{r}} \cdot \hat{\mathbf{v}})^n] \tag{6}$$

If $W(\theta)$ is set to the constant K_s , this equation becomes

$$I_p = I_a K_a + I_L [K_d (\hat{\mathbf{n}} \cdot \hat{\mathbf{I}}) + K_s (\hat{\mathbf{r}} \cdot \hat{\mathbf{v}})^n] \tag{7}$$

◇ Assignment 3

Put shading on the object you created in Assignment 1. Can you adjust the material property to adjust the intensities of diffuse light, specular light, and ambient light in your CAD software? Draw several pictures to show the effect of the adjustments. ◇

2. Shading Algorithms

Shading models are simple. **The input to a shading model is intensity and color of light source(s), surface characteristics at the point to be shaded, and the positions and orientations of surfaces and sources. The output from a shading model is an intensity value at the point.** Shading models are applicable to points only. To shade an object, a shading model is applied many times to many points on the object. These points are the pixels for a raster display. To compute a shade for each point on a 1024×1024 raster display, the shading model must be calculated over one million times. **These calculations can be reduced by taking advantage of shading coherence;** that is, **the intensity of adjacent pixels is either identical or very close.** There are several different shading algorithms.

2.1 Constant shading

This is the simplest and less realistic shading algorithm. **Since the unit normal vector of a polygon never changes, polygons will have just one shade.** An entire polygon has a single intensity value. **Constant shading makes the polygonal representation obvious and produces unsmooth shaded images (intensity discontinuities).**

2.2 Gourand shading

Gourand shading is a popular form of intensity interpolation or first-derivative shading. Gourand proposed a technique to eliminate (not completely) intensity discontinuities caused by constant shading. The first step in the Gourand algorithm is to calculate surface normals. If more than one polygon shares the same vertex as shown in Figure 4, the surface normals are averaged to give the vertex normal. If smooth shading between the four polygons shown is required, then

$$N_v = \frac{1}{4}(N_A + N_B + N_C + N_D) \quad (8)$$

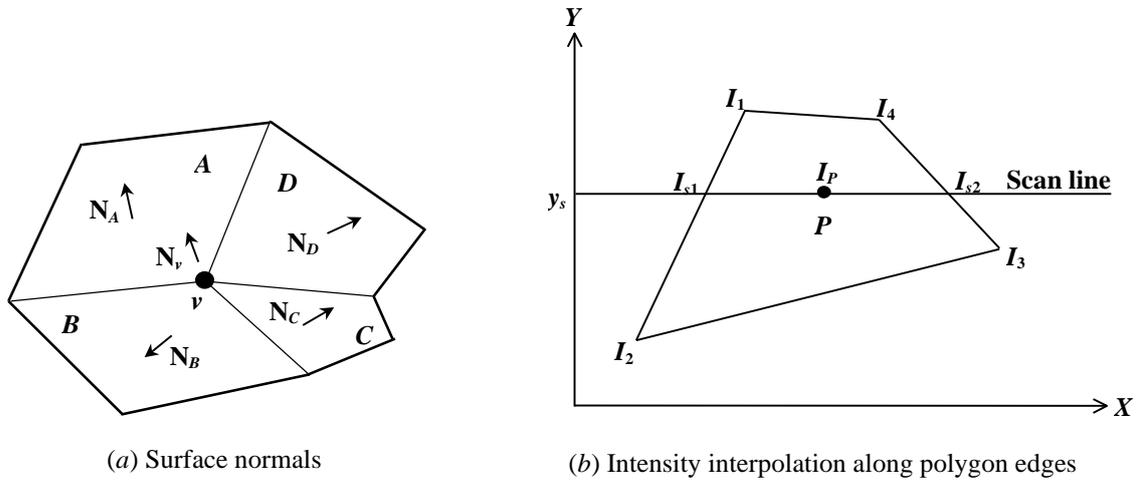


Figure 4. Gouraud shading.

The third step in the Gouraud algorithm (after calculating surface and vertex normals) is to compute vertex intensities using the vertex normals and the desired shading model [Equation (7)]. The fourth and the last step is to compute the shade of each polygon by linear interpolation of vertex intensities.

$$I_P = \frac{x_{s2} - x_p}{x_{s2} - x_{s1}} I_{s1} + \frac{x_p - x_{s1}}{x_{s2} - x_{s1}} I_{s2} \quad (9)$$

$$I_{s1} = \frac{y_s - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y_s}{y_1 - y_2} I_2 \quad (10)$$

$$I_{s2} = \frac{y_4 - y_s}{y_4 - y_3} I_3 + \frac{y_s - y_3}{y_4 - y_3} I_4 \quad (11)$$

Gouraud shading takes longer than constant shading and requires more planes of memory to get the smooth shading for each color. Figure 5 shows the Gouraud shading of a cone.



Figure 5. Gouraud shading of a cone

While Gouraud shading produces smooth shades, it has some disadvantages. If it is used to produce shaded animation (motion sequence), shading changes in a strange way because **interpolation is based on intensities and not surface normals** that actually change with motion. **Highlights are distorted due to the linear interpolation of vertex intensities.**

2.3 Phong shading

The basic idea behind Phong shading is to interpolate normal vectors at the vertices instead of the shade intensities and to apply the shading model [Equation (7)] at each point (pixel). To perform the interpolation, Equation (8) can be used to obtain an average normal vector at each vertex. Figure 6 shows phong shading of a cone.

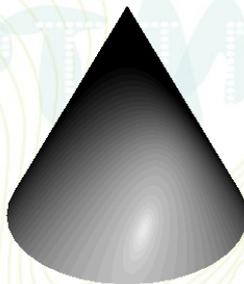


Figure 6. Phong shading of a cone

◇ Assignment 4

What are the shading algorithms provided in your CAD software? Use the object generated in Assignment 1 to illustrate different shading algorithms and discuss their differences. ◇

3. Shading Enhancements

The basic shading model is usually enhanced to produce special effects for both artistic value and realism purposes. These effects include **transparency, shadows, surface details, and texture**.

Two shading techniques can be identified: **opaque and translucent**. In the opaque technique, hidden surfaces in every pixel are completely removed. In the translucent method, hidden surfaces are not completely removed. This allows some of the back pixels to show through, producing a screen-door effect.

Consider the box shown in Figure 7. If the front face F_1 is made translucent, the back face F_2 can be seen through F_1 . The intensity at a pixel coincident with the locations of points P_1 and P_2 can be calculated as a weighted sum of the intensities at these two points, that is,

$$I = KI_1 + (1 - K)I_2 \quad (12)$$

where I_1 and I_2 are the intensities of the front and back faces respectively, calculated using, say, Equation (12). K is a constant that measures the transparency of the front face: when $K=0$, the face is perfectly transparent and does not change the intensity of the pixel; when $K=1$, the front face is opaque and transmits no light. Sometimes transparency is referred to as x-ray due to the similarity in effect.

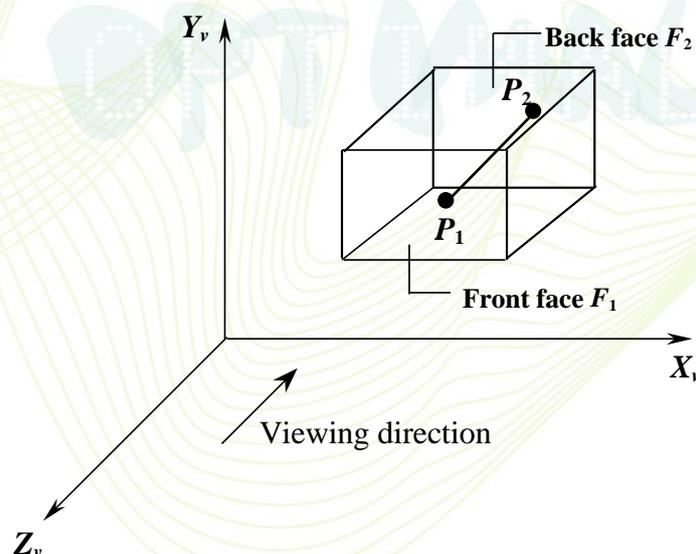


Figure 7. Transparency and visibility of back faces.

Shadows are important in conveying realism to computer images. More importantly, they facilitate the comprehension of spatial relationships between objects of one image. The complexity of a shadow algorithm is related to the model of the light source. If it is a point source outside the field of view at infinity, the problem is simplified. Finding which objects are in shadow is equivalent to solving the hidden surface problem as viewed from the light source. If several light sources exist in the scene, the hidden surface problem is solved several times—every time one of the light sources is considered as the viewing point. The surfaces that are visible to both the viewer and the light source are not shaded. Those surfaces that are visible to the viewer but not to the light source are shaded.

◇Assignment 5

Can you generate shadow with your CAD software? Can you generate shadows from multiple light sources? Use the object generated in Assignment 1 to demonstrate the functions in your CAD software. ◇

Texture is important to provide the illusion of reality. For example, modeling of a rough casting should include the rough texture nature of its surfaces. These objects, rich in high frequencies, could be modeled by many individual polygons, but as the number of polygons increases, they can easily overflow the modeling and display programs. **Texture mapping** (Figure 8) is introduced to solve this problem and provide the illusion of complexity at a reasonable cost. It is a method of “wallpapering” the existing polygons. As each pixel is shaded, its corresponding texture coordinates are obtained from the texture map, and a lookup is performed in a two-dimensional array of colors containing the texture. The value in this array is used as the color of the polygon at this pixel, thus providing the “wallpaper”.

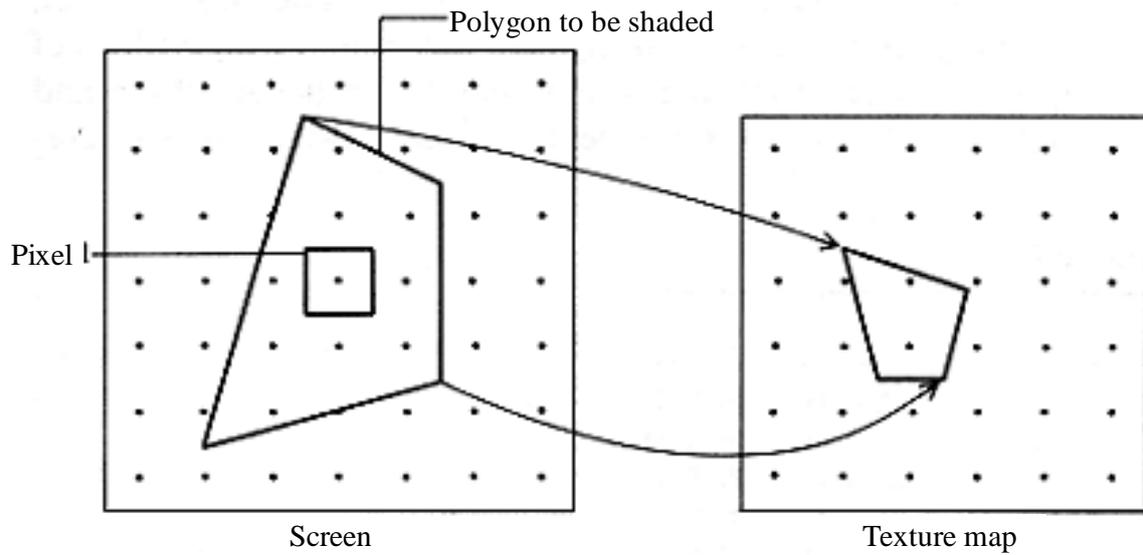


Figure 8. Texture mapping.

◇ Assignment 6

Use the object generated in Assignment 1 to demonstrate the texture mapping capability of your CAD software. Play around with the light sources and view points. Is there anything unnatural? ◇

