Rendering

HSR 3 and introduction to lighting models

z-buffer algorithm

A main aspect of this algorithm is that it does not need any preliminary ordering of the scene polygons. The algorithm efficiency is strongly affected by this fact.

Input/output The input to the algorithm is the unordered set of 3D polygons of the scene, already transformed in DC3 coordinates. The output is constituted by the final contents of the frame buffer, which contains a rasterized picture of the scene parts visible from the viewpoint.

Storage structures Two storage arrays are used, respectively denoted as *frame-buffer* and *z-buffer*. Such arrays have the same number of rows and columns, and their elements are associated to the pixels of the viewport on the display device. The frame-buffer contains a color value, usually either a composite RGB value or an index to some color look-up table. The z-buffer contains the discrete value of the z coordinate associated to the centroid of the polygon portion represented on a given pixel. Corresponding elements on the frame-buffer and on the z-buffer, say on the same row and column, are associated to the same polygon portion.

Notice that the z-buffer elements hold integer numbers, in a range of values depending on the number of bits per element, often equal to 24. Of course, the depth resolution of the algorithm depends on the quantity of RAM available for the z-buffer. This resolution may strongly affect the output quality when rastering 3D scenes of high geometric complexity.

Algorithm The algorithm initialization can be distinguished from the algorithm core. In the initialization step the frame-buffer is set to the background color, and the z-buffer is set to the minimum representable value, in the hypothesis that a right-handed frame is used for the DC3 coordinates. The view direction is that of positive z axis.

In the algorithm core the scene polygons are rasterized in any order. Let us consider the processing of the generic polygon. A covering with a discrete set of pixels at integer DC3 coordinates is produced, and the discrete z for each pixel is also computed. Such set of pixels give a minimal covering of the polygon image, but only a pixel subset is usually stored, according to their current visibility from the viewpoint.

Each polygon is rasterized by rows. For each generated pixel we need to decide if either it must be stored in the frame buffer and in the z-buffer, or not. Such a decision is positive if the pixel already stored in memory at the same xy address (i.e. at the same <column, row> pair) is further from the observer than the current pixel or vice versa. It is considered further when the integer value stored in the z-buffer is smaller than the z of the current pixel.

When all the scene polygons have been rasterized, the frame buffer will contain the scene picture really perceived from the viewpoint.

The generic rasterization step requires the computation of the z coordinate of the current pixel starting from the known z of the previous pixel.

Let $\boldsymbol{p}_k = (x_k, y_k)$ be the current pixel, belonging to the y_k row and to the x_k column of the device space. The z coordinate in $\boldsymbol{p}_k$ and $\boldsymbol{p}_{k+1}$ pixels may be computed by substituting the ordinate and abscissa of the two points into the plane equation of their polygon:

$$\begin{aligned} ax_k + by_k + cz_k + d &= 0, \\ ax_{k+1} + by_{k+1} + cz_{k+1} + d &= 0. \end{aligned}$$

In other terms:

$$\begin{aligned} z_k &= -\frac{1}{c}(d + ax_k + by_k), \\ z_{k+1} &= -\frac{1}{c}(d + ax_{k+1} + by_{k+1}), \end{aligned}$$

and hence

$$\Delta z = z_{k+1} - z_k = -\frac{a}{c}(x_{k+1} - x_k) - \frac{b}{c}(y_{k+1} - y_k).$$

But $y_{k+1} - y_k = 0$ and $x_{k+1} - x_k = 1$, so that

$$z_{k+1} = z_k + \Delta z = z_k - \frac{a}{c}. \tag{10.4}$$

$$z_{k+1} = z_k + \Delta z = z_k - \frac{a}{c}. \tag{10.4}$$

By comparing z_{k+1} with the value already stored in position (x_{k+1}, y_{k+1}) of z-buffer, it is possible to decide whether to store the pixel depth z_{k+1} or not. If the z value is stored, then the pixel color is also computed and stored. For this purpose other computations may be required, including normal interpolation and/or color shading, as discussed in the following sections.

Analysis It may be interesting to note that, as shown by equation (10.4), the computation of the z increment from the previous pixel only requires a subtraction by a constant number. This algorithm, which is extremely simple, is easily implemented in firmware on 3D graphics accelerators. Notice that the generation time of the hidden-surface removed picture depends linearly on the average area of polygons and on their number.

10.4 Illumination models

To generate realistic images by computer it is necessary both to remove the hidden-surfaces of the scene and to render the visible ones by taking into account the physical principles which regulate the diffusion and reflectance of light rays that incise on the external surfaces of bodies in the scene to be rendered. In particular, we need to take into account not only the geometry of the scene, but also some optical and physiologic aspects related to the diffusion, reflection and perception of light.

1. absorbed;
2. transmitted;
3. reflected, both diffusely and specularly,

in percentages that depend both on the body material and on the nature of the fixture of its exterior surface.

Lighting models The terms "lighting" or "illumination models" are used to denote the choice of some particular equation to compute the light intensity in a space point, as a function of the intensity of the incident radiation and the geometric and physical properties of the surface the point belongs to. The goal when using lighting models is to compute the light intensity on the surfaces of the scene, given some assigned distribution and configuration of light sources and some assigned material properties of the surfaces.

Sources When we model a scene, we have to give both its geometry and some suitable light sources. Such sources may be either set at infinity or at some finite position. As a consequence, light rays will be either parallel or divergent, respectively. Light sources may be geometrically shaped either as point sources or as distributed, i.e. extended, sources. Depending on requirements, we may have scenes with only one light source or with several sources. Also, we may imagine that the emitted light has the same intensity in all the directions, or that a source shines in a preferred direction. Such sources are called directional light sources.

Absorbed radiation The portion of incident radiation which is absorbed by a body is transformed into heat and produces an increase in temperature. Notice that if the body had absorbed all the incident light, then it would be not visible. Such perfectly absorbent body is called "black body". The absorbed radiation is not interesting for our purposes, but it may be useful to note that different bodies do not absorb the different wave lengths in the same way. For this reason the different materials look as if they have different colors when exposed to the daylight.

Transmitted radiation The incident radiation is partly transmitted inside the body when this is transparent. This case will not be considered in the remainder of the text. On this point we will only discuss how to set the percentage of transparency when stating the material properties in VRML.

Diffusely reflected radiation The diffuse radiation can be thought to be due to the portion of incident radiation which is absorbed by the external layer of the body, and then is uttered in all directions allowed by the position and orientation of the external surfaces. We will see that, at least in computing the diffuse intensity using the simplest diffusion model, it is not necessary to consider the viewpoint position, since the body, in the case of perfect light diffusion, behaves exactly as an emitter body.

Specularly reflected radiation We may imagine that the specularly reflected portion of the incident radiation does not traverse the external layer of the body, but is directly reflected in the plane defined by the direction of the normal to the surface in the considered point and by the direction of the incident light. In this case we need to compute how much reflected intensity may geometrically reach the viewpoint position.

10.4.1 Diffuse light

We know that the incident light on a opaque body may be partly diffused, partly reflected and partly absorbed by the body. Let us consider the simplest lighting model, where only the diffused light intensity is taken into account. The involved vector quantities are shown in Figure 10.26, where we have

1. $\boldsymbol{p}$ = point on the body surface;
2. $\boldsymbol{n}$ = normal unit vector to the surface in $\boldsymbol{p}$;
3. $\boldsymbol{\ell}$ = direction of incident light, given by the difference between the source position, supposed point-shaped, and $\boldsymbol{p}$;
4. θ = angle between vectors $\boldsymbol{\ell}$ and $\boldsymbol{n}$.

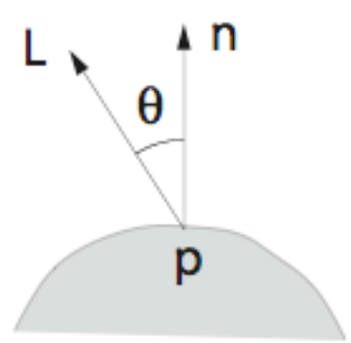

Figure 10.26 Lambert's cosine model of diffused reflection

Lambert's diffuse intensity The physical phenomenon is regulated by the following *Lambert's cosine law.* If a body is a perfect diffuser, then the diffused intensity $I_d(\boldsymbol{p})$ from a point $\boldsymbol{p}$ of its surface is proportional to the incident light intensity and to the cosine of the angle θ:

$$I_d(\boldsymbol{p}) = K_d I_\ell \cos\theta, \tag{10.5}$$

where I_ℓ is the light intensity incident in $\boldsymbol{p}$. The factor K_d is called *diffusion constant.* The θ angle must necessarily vary between 0 and $\frac{\pi}{2}$.

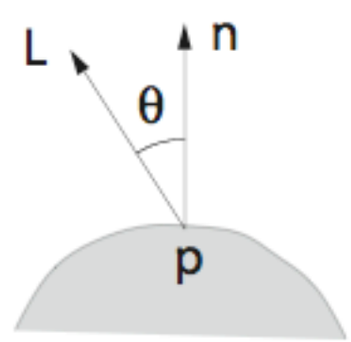

Figure 10.26 Lambert's cosine model of diffused reflection

The model described by equation 10.5 is the simplest lighting model, with a single light source which illuminates a single opaque body. For θ greater than $\frac{\pi}{2}$, we have $I_d(\boldsymbol{p}) = 0$, where no light would be available since the $\boldsymbol{p}$ point would not be visible from the source.

Such a single assumption is too strong, because it would render as *black* all the surface portions where $\frac{\pi}{2} \leq \theta$. In other words, when using this simple model, we should render as black all the scene polygons which are not presently visible from both the viewpoint and the light source. Such a result would be actually unacceptable.

Ambient diffuse intensity The problem of black surfaces discussed above may be easily solved by considering that in any scene there is some diffused lighting due to surrounding environment. For this purpose it is sufficient to assume that on each body surface there is also some, usually small, quantity of light not directly received from a light source but coming from the ambient surroundings of the scene. In other terms, it is assumed that each body will diffuse a constant portion of the ambient lighting.

An additive *ambient* term $K_a I_a$ is hence introduced in the diffusion equation 10.5, depending on the average *ambient lighting* I_a, characteristic of the ambient of the scene, and on the *ambient diffusion* constant K_a, characteristic of the body material:

$$I_d(\boldsymbol{p}) = K_d I_\ell \cos\theta + K_a I_a \tag{10.6}$$

Notice that black surfaces may appear again, as soon as either K_a or I_a are equal to zero. But also this simple model has some important drawbacks. Consider in fact two parallel polygons made with the same material and located at different distances from the viewpoint. By using the above equation they would have the some light intensity in each point (and hence exactly the same color), so that, in case of partial visual occlusion with respect to the viewpoint, it would be actually impossible for the observer to distinguish the first surface from the second surface.

Depth attenuation of intensity To get more realistic results it is necessary to take into account the intensity attenuation as a function of depth. In particular, we should consider as appropriate physical parameter the *intensity flow*, defined as the energy which crosses the unit surface. This parameter would be measured at distance d by the intensity diffused in $\boldsymbol{p}$, over the area $4\pi d^2$ of the spherical surface centered in $\boldsymbol{p}$. The intensity attenuation should be hence made proportional to the inverse of the squared distance.

But dividing the first term of equation (10.6) by the square of the viewpoint distance gives results which do not experimentally match the user experience. In particular it is possible to notice that by varying the intensity (i.e. the color) with the inverse of the square of the viewpoint distance produces a too strong variation between points belonging to surfaces located at relatively small distances from each other.

Conversely, some good visual results are given by the equation:

$$I_d(\boldsymbol{p}) = \frac{K_d}{d+K} I_\ell \cos\theta + K_a I_a \tag{10.7}$$

where d is the viewpoint distance from $\boldsymbol{p}$ and K is an additive constant which gives a "fine-tuning" of the lighting behavior of surfaces, until to make them look realistic. Notice that d cannot be interpreted as the distance from viewpoint if this one is at infinity. In such a case d is assumed as the z difference between $\boldsymbol{p}$ and the scene point with maximum z.

10.4.2 Specular reflection

If a body is perfectly reflective, say, a perfect mirror, then all the radiation incident on $\boldsymbol{p}$, which is not absorbed or transmitted, is reflected along the symmetric direction of $\boldsymbol{\ell}$ with respect to $\boldsymbol{n}$, i.e. along the reflection vector $\boldsymbol{r}$. See Figure 10.27.

With a perfectly reflective body, the observer would see the considered point only if located along the reflection direction. Actually, real bodies do not behave like perfect ones, and do not reflect the light only along the reflection direction. Conversely, the reflected intensity is spatially distributed around $\boldsymbol{r}$. Hence the viewer, located on the view direction $\boldsymbol{v}$, may in any case get a perception of some portion of the reflected light, and can see $\boldsymbol{p}$ when $\theta + \alpha < \frac{\pi}{2}$.

In particular, the reflected light intensity perceived along $\boldsymbol{v}$ is a function of the angle α between $\boldsymbol{v}$ and $\boldsymbol{r}$, as shown in Figure 10.27. Notice that the view direction $\boldsymbol{v}$ is not necessarily coplanar with the $\boldsymbol{\ell}$, $\boldsymbol{n}$ and $\boldsymbol{r}$ vectors.

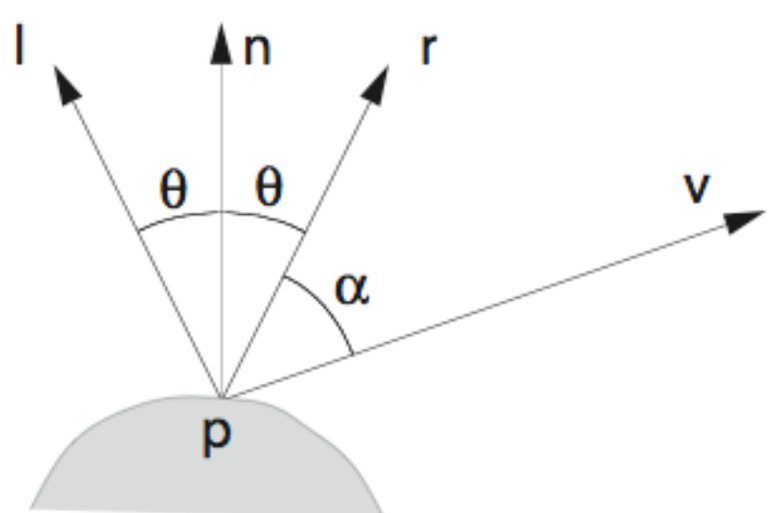

Figure 10.27 Specular reflection from a point source

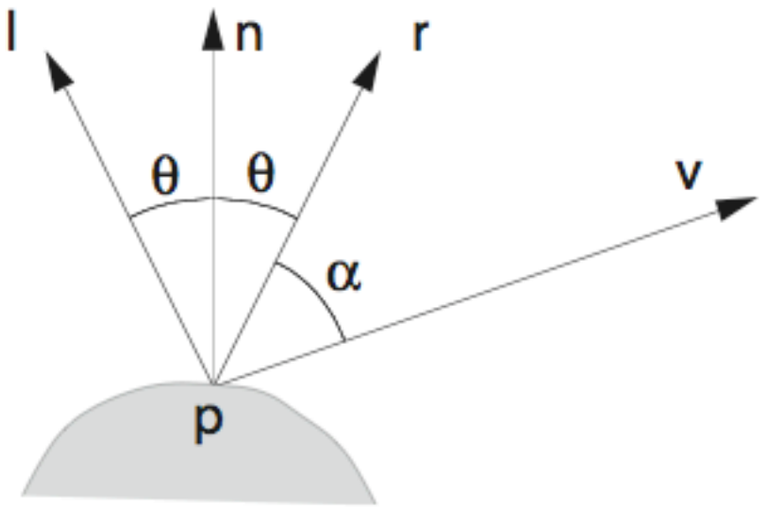

Figure 10.27 Specular reflection from a point source

The quantity of specularly reflected energy strongly depends in a quite complex way both on the wavelength λ of the incident light and on the angle α between the reflection and view directions.

Phong's reflection model A simplified description of the phenomenon is due to Phong Bui-Tuong [Pho75]. In this model it is assumed that the *specularly reflected intensity* $L_s(\boldsymbol{p})$ is proportional to (a) the incident radiation I_ℓ as well as to (b) the n-th power of cosine of angle α between the reflection and the view directions, through a function $w(\theta, \lambda)$ which depends on the incidence angle θ and on the wavelength λ.

$$L_s(\boldsymbol{p}) = w(\theta, \lambda) I_\ell \cos^n \alpha$$

The n parameter depends on the body material. In Figure 10.28 we show the graph of $\cos^n \alpha$ function, as generated by Script `PLaSM` 10.4.1.

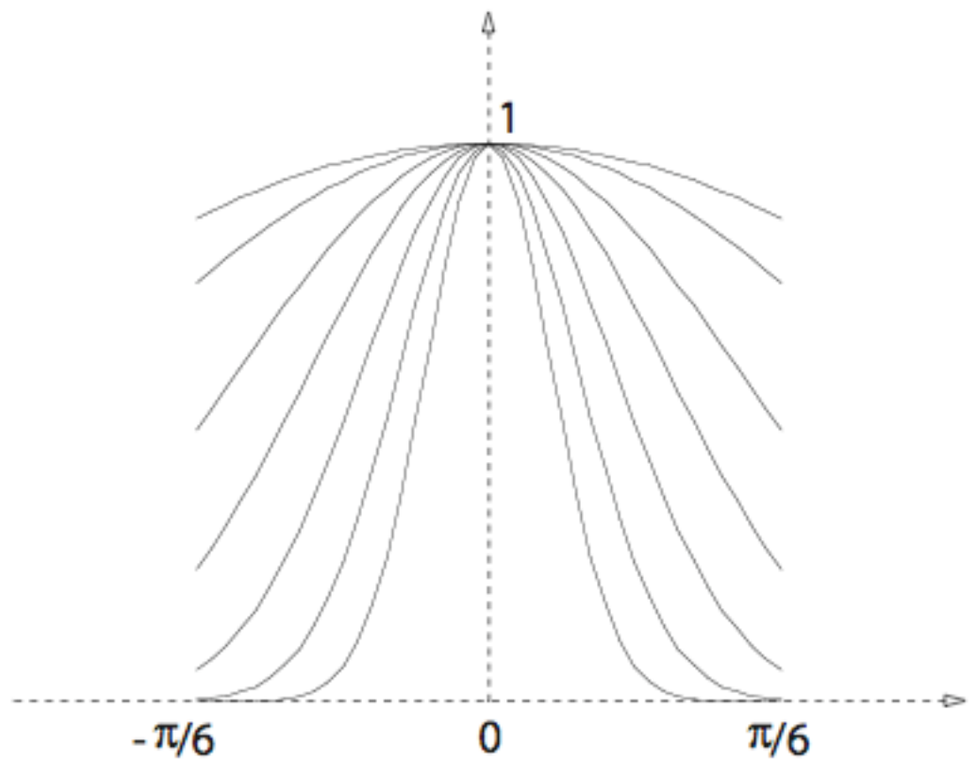

Figure 10.28 Graph of function $\cos^n \alpha$, for $-\frac{\pi}{6} \leq \alpha \leq \frac{\pi}{6}$, and with samples of n between 1 and 80

Simplified model of specular reflection The w function of Phong's model, which takes into account both the incidence angle θ and the incident wavelength λ, is actually very complex, and can only be given empirically. It is usually substituted by a constant K_s, called the *specular reflection constant*, so that we have the simplified reflection model:

$$L_s(\boldsymbol{p}) = K_s I_\ell \cos^n \alpha.$$

The material properties are thus incorporated in the parameter n, whereas the geometric (incidence angle) and physical (wavelength) properties of the incident light are summarized by the K_s constant. Since the aspects already discussed for the diffused intensity continue to hold, the following better expression is usually adopted for the specularly reflected intensity:

$$I_s(\boldsymbol{p}) = \frac{K_s}{d + K} I_\ell \cos^n \alpha + K_a I_a. \tag{10.8}$$

Aggregated reflection models In summary, the expressions already seen for diffuse and specularly reflected intensities can be aggregated, since common bodies usually behave both as light diffusers and as mirrors. The aggregated model for the reflected light intensity is

$$I(\boldsymbol{p}) = \frac{I_\ell}{d+K}(K_d \cos\theta + K_s \cos^n \alpha) + K_a I_a \tag{10.9}$$

where K_d, K_s and n are material properties, I_ℓ depends on the light source, and α, θ, K and d are characteristics of the geometric configuration of the surface. The geometric parameters of the illumination model are better "highlighted" (!) by using inner products of unit vectors rather than cosines of angles:

$$I(\boldsymbol{p}) = \frac{I_\ell}{d+K}\left(K_d \boldsymbol{\ell} \cdot \boldsymbol{n} + K_s \left(\boldsymbol{r} \cdot \boldsymbol{v}\right)^n\right) + K_a I_a. \tag{10.10}$$

More in general, several light sources may appear on the stage, with different incident light intensities and properties. Hence, for the reflected intensity $I(\boldsymbol{p})$ in a point illuminated by various sources we can write

$$I(\boldsymbol{p}) = \sum_i \left[\frac{I_{\ell_i}}{d + K_i} (K_d \, \boldsymbol{\ell}_i \cdot \boldsymbol{n} + K_s \, (\boldsymbol{r}_i \cdot \boldsymbol{v})^n) \right] + K_a I_a.$$

Geometry of reflection models Vectors $\boldsymbol{\ell}, \boldsymbol{n}$ and $\boldsymbol{v}$ are independent variables of the geometric problem. In particular, $\boldsymbol{\ell}$ can be obtained by normalizing the point difference between the source location and the considered point $\boldsymbol{p}$. The $\boldsymbol{v}$ vector is computed analogously, by normalizing the point difference between the viewpoint and $\boldsymbol{p}$.

The normal vector $\boldsymbol{n}$ is constant on each planar surface. In a polygon it can be computed by the vector product of two consecutive edges with an internal angle less than $\frac{\pi}{2}$. Such a convex internal angle always exists, for both convex and concave polygons, where the common vertex of two edges is extremum with respect to one coordinate, e.g. is the point with either maximal or minimal z, provided that the polygon does not lie on a plane with equation $z = c$.

The computation of the direction of reflection $\boldsymbol{r}$ is a bit more complex. In this case it is necessary to compute a rototranslation tensor $\boldsymbol{Q}$ which maps $\boldsymbol{p}$ in the origin and the normal unit vector $\boldsymbol{n}$ in the basis vector $\boldsymbol{e}_3$ of the z axis. So we have, in homogeneous coordinates:

$$\boldsymbol{r} = \boldsymbol{Q}^{-1} \boldsymbol{S}_{xy}(-1, -1)\, \boldsymbol{Q}\, \boldsymbol{\ell},$$

where $\boldsymbol{S}_{xy}(-1, -1)$ is the scaling tensor that reverses the sign of the x and y coordinates.