

# Illumination Models and Surface Rendering Methods

Realistic displays of a scene are obtained by perspective projection and applying natural light effects to the visible surfaces an illumination model (lighting model) and sometimes called shading model, is used to calculate the intensity of light that we should see at a given point on the surface of an object.

A surface rendering algorithm uses the intensity calculations from an illumination model to determine the light intensity calculation from an illumination model.

## Light sources

Point source: tungsten filament bulb image can be seen

Distributed light source: fluorescent light

When light is incident on an opaque surface part reflected part absorbed. Surface that are rough or grainy tend to scatter reflected light in all direction is called diffuse reflection

Light sources create highlights or bright spots called specular-reflection.

Ambient light surface directly not exposed directly but visible if nearby objects are illuminated

Combination of light reflections from various surfaces to produce a uniform illumination called the ambient light or background light(no shadow's produced)

It has no spatial or direction characteristics and amount on each object is a constant for all surfaces and over all directions

## Diffuse reflection

Ambient light is an approximation of global diffuse, light effects .

Diffuse reflections are constant over each surface in a scene independent of viewing direction

$k_d$  or diffuse reflection coefficient or diffuse reflectivity (0 to 1)

$k_d$  is nearly 1 for highly reflective surface and  $k_d$  is 0 where light absorbs (black surfaces)

Diffuse reflection intensity at any point on the surface as

$$I_{\text{ambDiff}} = k_d \cdot I_d$$

Where  $I_{\text{ambDiff}}$  is ambient light due to diffusion and  $I_d$  is light due to diffusion assuming diffuse reflections from the surface are scattered with equal intensity in all directions independent of the

Viewing direction (called “ideal diffuse reflectors”) also called Lambertian reflectors and governed by Lambert’s Cosine Law.

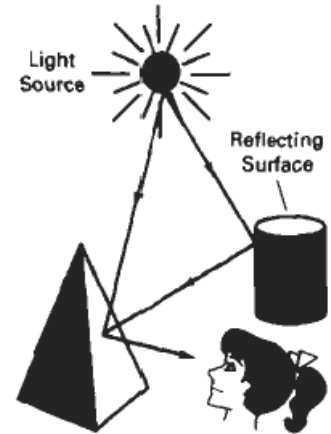
If “angle of incidence” between incoming light direction and surface normal is  $\theta$

$$I_{\text{LDiff}} = k_d \cdot I_L \cos \theta$$

where  $I_{\text{LDiff}}$  is light due to diffusion

If  $N$  is unit normal vector to a surface and  $L$  is unit direction vector to the point light source then

$$I_{\text{LDiff}} = k_d \cdot I_L (N \cdot L)$$



In addition many graphics packages introduce an ambient reflection coefficient  $k_d$  to modify ambient light intensity  $k_d$

then 
$$I_{\text{Diff(i.e. total)}} = k_a \cdot I_a + k_d \cdot I_L (N \cdot L)$$

### Assigning Intensity Levels

To display  $n+1$  successive intensity levels with equal perceived brightness the intensity levels on the monitor should be spaced so that ratio of successive intensities is constant.

$$I_1 / I_0 = I_2 / I_1 = \dots = I_n / I_{n-1} = r \quad (0 < \text{intensity} < 1)$$

$I_0$  = lowest intensity level and  $I_n$  = highest intensity level

any intermediate intensity in term of  $I_0$   $I_k = r^k I_0$  to calculate  $r$  substituting  $k = n$  and  $I_k = I_n = 1$

$$r = (1/I_0)^{1/n}$$

e.g. if  $I_0 = 1/8$  and  $n = 3$ ,  $r = 2$  four intensities are  $1/8$   $1/4$   $1/2$  and 1

Lowest intensity value  $I_0$  depends on the characteristics of the monitor typically from 0.005 to 0.025

For B.W. monitor with 8 bits per pixel ( $n = 255$ ) and

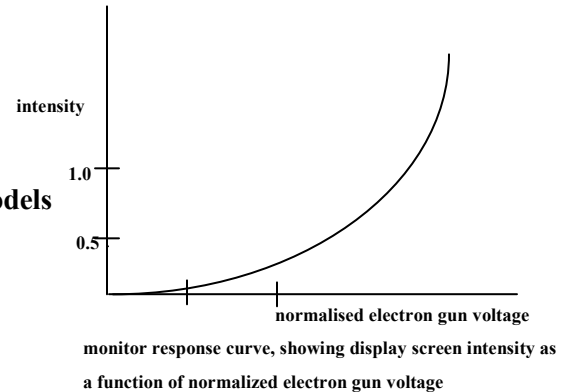
$I_0 = 0.01$   $r = 1.0182$  and approximate values for the

256 intensities on the system are 0.01, 0.0102,

0.0104,....., 0.98211, and 1.00.

For color, we setup intensity levels for each component of the color models

Blue component of RGB  $I_{Bk} = r_B^k I_{B0}$  where  $r_B = \left( \frac{1}{I_{B0}} \right)^{1/n}$

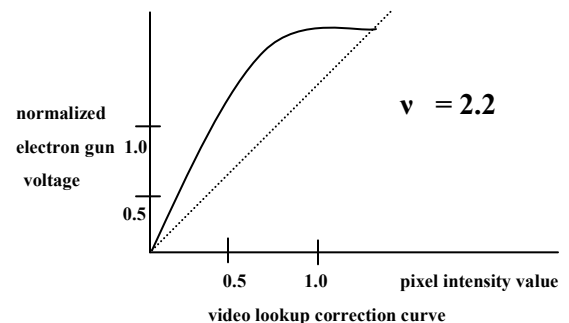


### Gamma Correction and Video Look up Table

Problem associated with the display of calculated intensities is the non linearity of display devices and illumination model produce a linear range of intensities.

Usually calculated intensities are stored in an image file as integer values (1 byte for each of three RGB components)

If we set the voltages for the electron gun to the linear pixel values the displayed intensities will be shifted according to the monitor response curve



To correct for monitor nonlinearity graphics systems use a video lookup table that adjusts the linear pixel values  
Monitor response curve is described by the exponential function

$$I = a V^v \quad \text{and} \quad V = \left( \frac{I}{a} \right)^{1/v} \quad \text{where, } I = \text{displayed Intensity, } V = \text{input parameter}$$

$a$  and  $v$  depends on monitor characteristics

This calculation is referred to as “Gamma Correction of Intensity”,  $v \rightarrow 2$  to 3 for NTSC  $v = 2.2$

We combine gamma correction with logarithmic intensity mapping to produce a lookup table that contains both conversions.

### Intensity Attenuation (I A)

Intensity is attenuated by the factor  $1/d^2$  ( $d$  – distance that light has traveled)

Graphics package have a general inverse quadratic attenuation function

$$F(d) = \frac{1}{a_0 + a_1 d^1 + a_2 d^2}$$

User can fiddle with  $a_0, a_1, a_2$  to obtain a variety of lighting effects for a scene

Basic illumination model is modified as

$$I = k_a \cdot I_a + \sum_{i=1}^n f_i(d_i) I_{Li} [k_d (N \cdot L_i) + K_s (N \cdot H_i)^S]$$

### Color Considerations

Diffuse reflection coefficient vector for RGB component ( $K_{dR}, K_{dG}, K_{dB}$ )

For blue reflectivity component ( $K_{dR} = K_{dG} = 0$ )

$$I_B = k_{aB} \cdot I_{aB} + \sum_{i=1}^n f_i(d) I_{LBi} [k_{dB} (N \cdot L_i) + K_{sB} (N \cdot H_i)^S]$$

Another method diffuse and specular color vector

$$I_B = k_a S_{dB} \cdot I_{aB} + \sum_{i=1}^n f_i(d) I_{LBi} [k_d S_{dB} (N \cdot L_i) + K_s S_{sB} (N \cdot H_i)^S]$$

Color specification with its spectral wavelength

$$I = k_a S_d \cdot I_a + \sum_{i=1}^n f_i(d) I_{Li} [k_d S_d (N \cdot L_i) + K_s S_s (N \cdot H_i)^S]$$

### Transparency

A transparent surface in general produces both reflected and transmitted light

$\theta_r$  = angle of refraction  $n_r$  - index of refraction

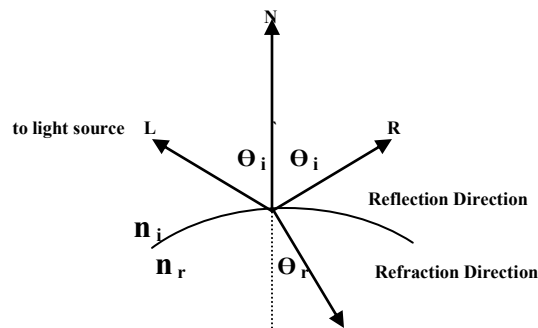
$\theta_i$  = angle of incidence  $n_i$  - index of incidence

using Snell's law

$$\sin \theta_r = \frac{n_r \sin \theta_i}{n_i}$$

Also,  $I = (1 - k_t)I_{\text{reflect}} + k_t I_{\text{trans}}$

Where,  $I_{\text{trans}}$  is transmitted intensity  
 $I_{\text{reflect}}$  is reflected intensity  
 $k_t$  transparency coefficient  
 $(1 - k_t)$  is the opacity factor



For highly transparent object  $k_t$  is near 1

For opaque objects  $k_t$  is near 0

### Shadows

Hidden surface method can be used to locate area where light sources produce shadows

Once we have determined the shadow area for all light sources, the shadows could be treated as surface patterns and store in pattern arrays