Color, Illumination Models, and Shading

by William Shoaff with lots of help

January 31, 2002

Overview

1. PDF version of these notes
2. Audio file of these notes (Windows Media Audio [wma])

1 Rendering

Webster's dictionary defines "to render" as "to reproduce or represent by artistic means." We want to render scenes using the medium of computer graphics. The basic topics to be covered here are: color, illumination, and shading.

2 Color

The subject of color is too complex to cover completely, but we can cover it adequately. To really understand what color is all about, we'd need classes in physics, physiology, psychology and art. Interpreted color of an object depends on at least the following factors.

- The object's surface – is it rough, smooth, shiny, dull?
- The light sources – are they points, lines, regions; where are they? what wavelengths do they emit?
- The surrounding (ambient) environment – are we outside or inside?
- The human visual system – how's are color vision? what's our perceived range of colors?

Before we delve into some of these topics let's consider "uncolored" light.

2.1 Achromatic Light

Achromatic light runs the gamut from black through grays to white light. The *quantity* of light is the only attribute of achromatic light. Physicists talk of *intensity* or *luminance* of the light energy. Psychologists talk of the *brightness* of the *perceived* intensity. Luminance and brightness are related by not identical concepts. We'll scale these quantities so that

- Zero (0) is defined as black.
- One (1) is defined as white.
- Gray is any value between 0 and 1.

Gray scale display devices can produce values 0, 1, and some discrete subset between 0 and 1. Monochrome or bi-level display devices produce only black and white images.

2.1.1 Gamma Correction of Cathode Ray Tubes (CRTs)

The eye is sensitive to ratios of intensity levels rather than absolute intensity. For example, we perceive changes in intensity from 0.1 to 0.11 as equal to a change in intensity from 0.5 to 0.55. The ratios are equal: 0.11:0.1 = 0.55:0.5. One the other hand, when we switch a three way bulb from 50 to 100 to 150 watts, our perception of the change in intensity is not identical: Switching from off to 50 watts is an infinite increase in brightness, while Switching to 50 from 100 seems to give is less of an increase and the change from 10 to 150 watts is even less of an increase in brightness. These are physiological phenomena.

What does this mean for computer graphics? We'll suppose that you had a gray scale device with settings of 0, 1/4, 1/2, 3/4, and 1. If these values corresponded exactly to the intensity or brightness of the display we would not perceive them as uniform changes in brightness. The ratios of consecutive values: ∞, 2, 3/2, and 4/3 are not uniform. We need to "correct" the values to obtain uniform ratios for uniform gray levels.

Good modern graphics monitors will compensate for our non-linear perception of intensity changes by *gamma correction*. For those that don't we should implement gamma correction in software. Here's how it works.

To select $n+1$ intensities I_0, $I_1, \ldots, I_n$ between 0 and 1 so that they have equal steps in brightness (perceived intensity) we'll find a fixed ratio r such that $r = I_{j+1}/I_j$ for all $j = 0,\ 1, \ldots, n-1$. That is,

$$I_0 = I_0,\ I_1 = rI_0,\ I_2 = rI_1 = r^2 I_0, \ldots,\ I_n = rI_{n-1} = r^n I_0 = 1.$$

Using the last setting $r^n I_0 = 1$, we find

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

and

$$I_j = r^j I_0 = I_0^{(n-j)/(n)}, \quad j = 1, \ldots, n.$$

The lowest intensity level depends on the display. Typical values for I_0 are between 0.005 and 0.025. Notice that perfect black is not possible. The *dynamic range* of a CRT is the ratio between maximum and minimum intensities, that is the dynamic range is $I_n/I_0 = 1/I_0$.

The display of an intensity I_j is still tricky. Given that there are N electrons in a beam hitting a phosphor, the intensity of light output is determined by the formula

$$I = kN^{\gamma},$$

where k and γ are constants, with γ typically in the range 2.2 to 2.5. The number of electrons N is proportional to the control grid voltage, which is proportional to the value V specified for the pixel. That is,

$$I = KV^{\gamma}, \quad \text{or} \quad V = (I/K)^{1/\gamma},$$

for some constant K. Given I, we calculate the nearest intensity I_j by

$$I_j = r^j I_0, \quad \text{where} \quad j = \lfloor \log_r (I/I_0) + 0.5 \rfloor.$$

And, the pixel value is

$$V_j = \lfloor (I_j/K)^{1/\gamma} + 0.5 \rfloor$$

This value, V_j, is placed in the frame buffer, or j is placed in the frame buffer and V_j in entry j of the color look-up table.

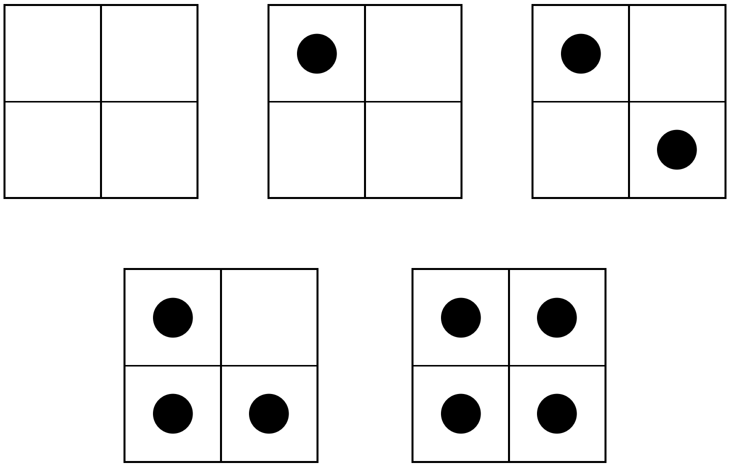


Figure 1: Five intensity levels for a 2×2 grid.

2.1.2 Halftone Approximation

Halftoning is a method used to represent shades of gray on a strictly monochrome (black and white) device, for example, black and white newspaper photographs are halftone images that appear to have shades of gray. The idea is to use multiple pixels (dots) to represent a single pixel (dot).

Suppose an image has fewer pixels than the device on which it will be displayed. Then perceived intensity range of the image can be increased, at a cost of reduced spatial resolution, by *halftoning* or *clustered-dot ordered dithering*, which uses variable sized black circles to produce different intensity levels. Graphics devices can approximate variable-area circles. For example, with a 2×2 grid, 5 different intensity levels can be achieved, as shown in figure 1

It is easy to see that on an $n \times n$ grid $n^2 + 1$ intensity levels can be produces as 0, 1, 2, …, n^2 pixels are lit. Of course, the question remains: Which pixels should be lit?

There are several guidelines for selecting the set of patterns for halftoning. Most of them are subjective.

- Avoid patterns that will introduce visual artifact, eg. horizontal, vertical, or diagonal line, in the image
- Form a *growth sequence* with the patterns — any pixel intensified at level j is intensified at level k for $k > j$
- Grow the patterns outward from the center, to create effect of increasing dot size
- For devices that can't plot isolated dots well, eg. laser printers, all "on" pixels should be adjacent

Dither matrices can be used to represent a sequence of patterns, e.g.

$$D^{(2)} = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix}, \quad D^{(3)} = \begin{bmatrix} 6 & 8 & 4 \\ 1 & 0 & 3 \\ 5 & 2 & 7 \end{bmatrix}$$

which show the order in which pixels are lit to increase the intensity.

Halftone approximation can also be applied to gray-scale and color devices. Consider a gray-scale device with With m bits/pixel. It can generate 2^m intensities, which we will represent as $0,\, 1, \ldots,\, 2^m - 1$. Using an $n \times n$ grid for halftoning,

$$n^2 * (2^m - 1) + 1$$

intensities from 0 to $n^2 * (2^m - 1)$ can be formed. On an RGB color device with m_R, m_G, and m_B bits for red, green, and blue respectively,

$$(n^2 * (2^{m_R} - 1) + 1) * (n^2 * (2^{m_G} - 1) + 1) * (n^2 * (2^{m_B} - 1) + 1)$$

colors can be formed.

Dispersed-dot ordered dithering can be used on devices able to display individual dots. When the device and the image have the same number of pixels, a pixel at $(x,\, y)$ can be intensified if the intensity $S(x,\, y)$ is greater than the value in the dither matrix at the entry $(i,\, j)$ corresponding to $(x,\, y)$, ie. if

$$S(x,\, y) > D^{(n)}_{ij}, \text{ where } i = x \bmod n,\, j = y \bmod n$$

The effect of equal image and display arrays is apparent only in areas where intensity varies

2.2 Chromatic Light

Now we want to consider color in more detail. Color *perception* involves three attributes: *hue*, *saturation*, and *lightness*. *Hue* or *chromaticity* classifies the *quality* of the color as determined by its dominant wavelength. You use color names such as red, yellow, orange, green, cyan, blue, magenta, etc., to specify a hue. *Saturation* or *chroma* is the degree of difference from the achromatic light of the same brightness. It is the chromatic purity: freedom from dilution with white and hence vividness of hue. A saturated color is a pure *spectral* color, an unsaturated color is pastel. *Lightness* is the perceived intensity of a *reflecting* object, while *brightness* is the perceived intensity of a *self-luminous* object.

Colorimetry is a branch of physics which provides an objective, quantitative way of measuring color. The basics include measurements of:

- *dominant wavelength* of light: This corresponds to the perceptual notion of hue.
- *excitation purity* of light: This corresponds to saturation.
- *luminance* measures intensity or energy of light: This corresponds to lightness or brightness.

Visible light is electromagnetic energy in the range of about 400 to 700 nanometers in wavelength (λ). The hues we perceive range from violet to indigo to blue to green to yellow to orange to red as wavelength goes from 400 to 700 nanometers. The human eye can distinguish hundreds of thousands of colors. Near the ends of the spectrum colors of noticeably different hues may be about 10 nm apart. Most distinguishable hues are within 4 nm of each other. About 128 fully saturated hues can be distinguished

Light can also be described by frequency f which is related to wavelength λ by

$$f = \frac{c}{\lambda},$$

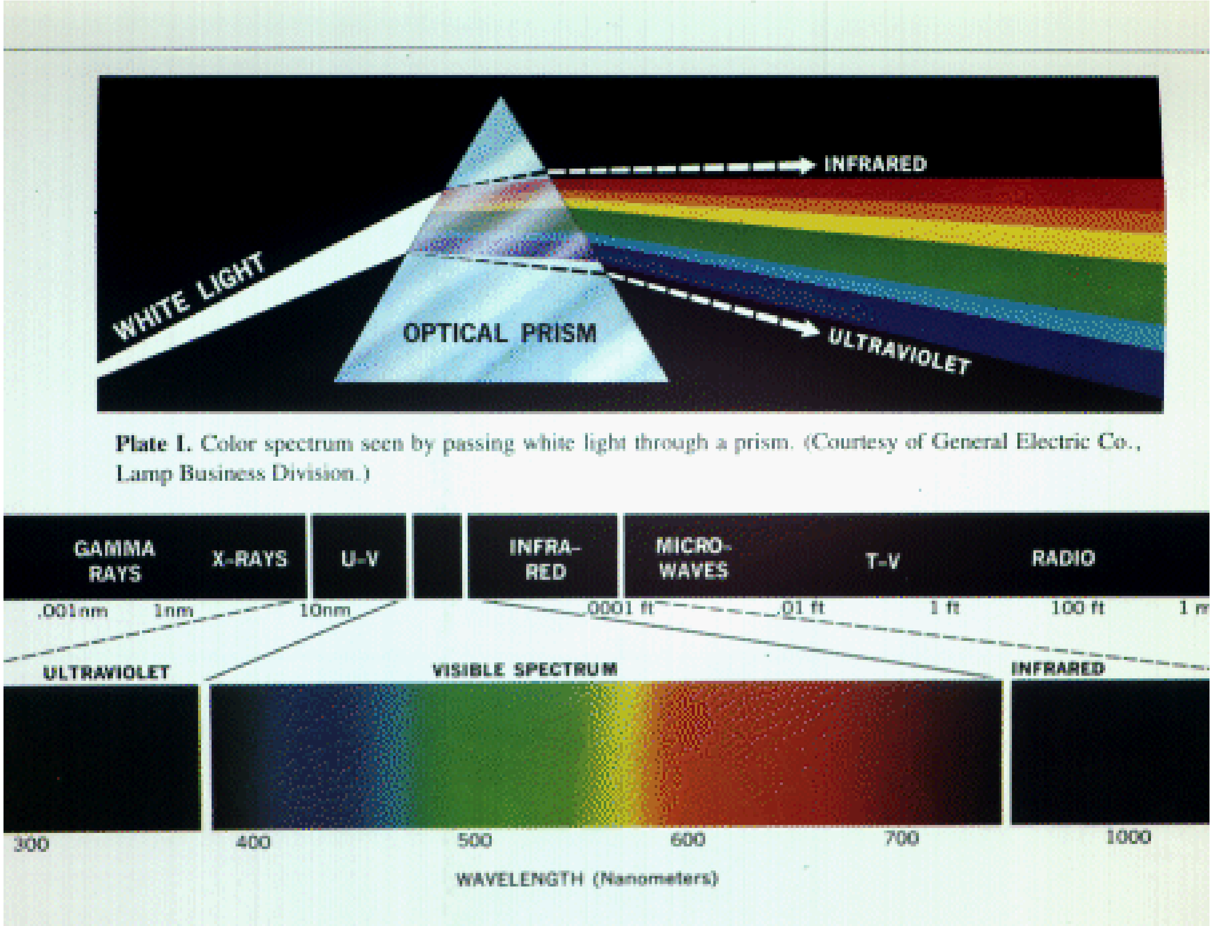


Plate 1. Color spectrum seen by passing white light through a prism. (Courtesy of General Electric Co., Lamp Business Division.)

Figure 2: The visible spectrum.

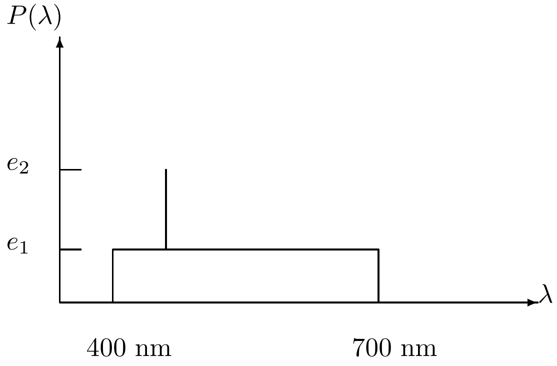


Figure 3: Idealize spectral energy distribution.

where $c \approx 300,000$ kilometers per second is the speed of light in a vacuum. The *spectral energy distribution* $P(\lambda)$ of light gives the amount of energy present at each wavelength λ. Two spectral energy distributions that “look” the same are called *metamers*.

We want to relate qualities of the spectral energy distribution to colorimetry and perception.

The dominant wavelength λ_d is where the largest spike e_d in the spectral energy distribution occurs The excitation purity is the percentage difference between e_d and the uniform distribution of energy The luminance is proportional to the area under the spectral energy distribution curve, weighted by the *luminous-efficiency function*

2.2.1 The tristimulus theory

The human visual system plays an integral part in the theory of color. Within the human eye are two element which are responsible for the perception of light. These are the rods and cones. The rods contain the elements that are sensitive to light intensities. They are used almost exclusively at night for humans night vision. The cones provide humans with vision during the daylight and are believed to separated into three types, where each type is more sensitive to a particular wavelength.

The perception of color by the human visual system is based on the *tristimulus theory*, which states that color vision results from the action of three types of cone receptor mechanisms with different spectral sensitives. When light of a particular wavelength is presented to the eye, these mechanisms are stimulated to different degrees, and the ratio of activity in the three mechanisms results in the perception of a color. Each color is therefore, coded in the nervous system by its own ratio of activity in the three receptor mechanism. An estimate of the eye’s response to these spectral sensitivities is shown in figure 6.

Experiments have produced *spectral-response functions* which show

- blue has peak response around 440 nm, with about 2% of light absorbed by the blue cones;
- green has peak response around 545 nm, with about 20% of light absorbed by the green cones;
- red has peak response around 580 nm, with about 19% of light absorbed by the red cones;
- the eye’s response to blue light is much less strong than it is to green or red;

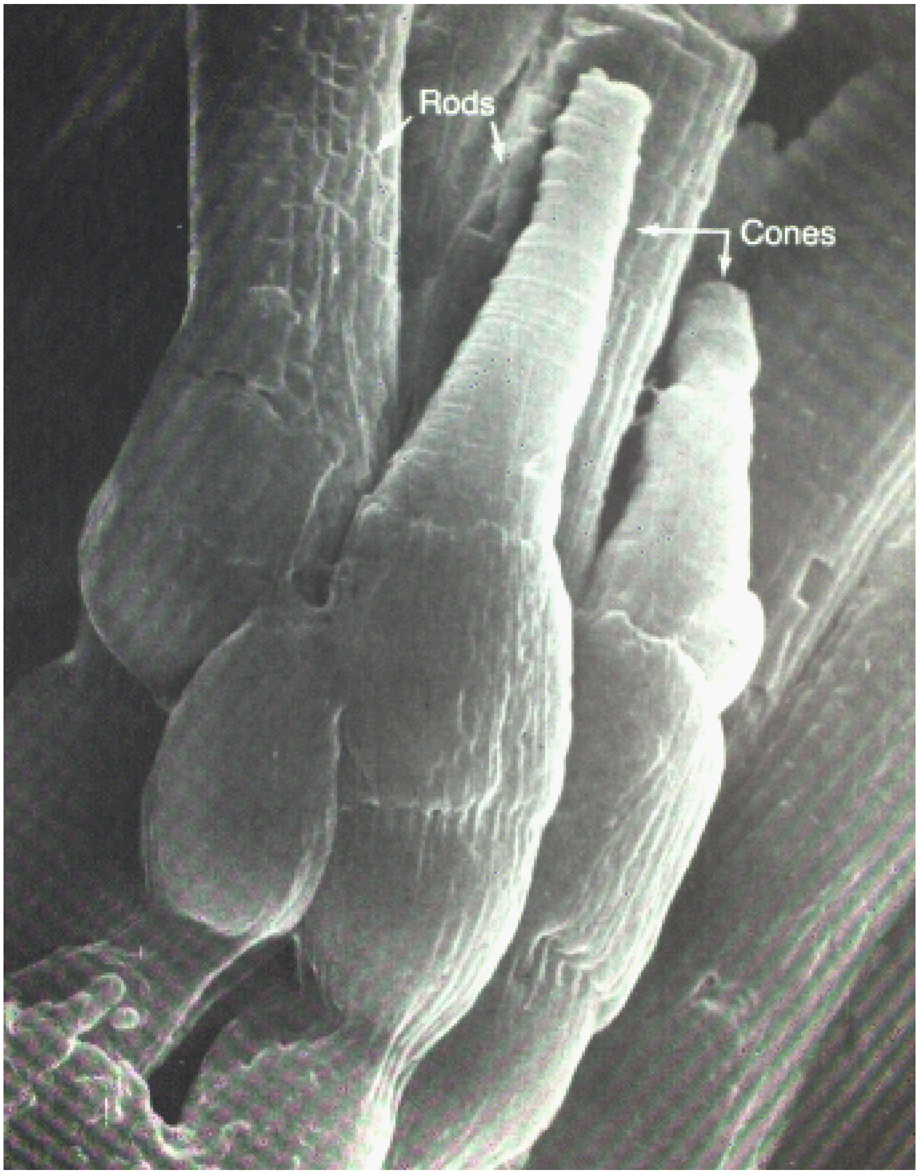


Figure 4: Photograph of rods and cones.

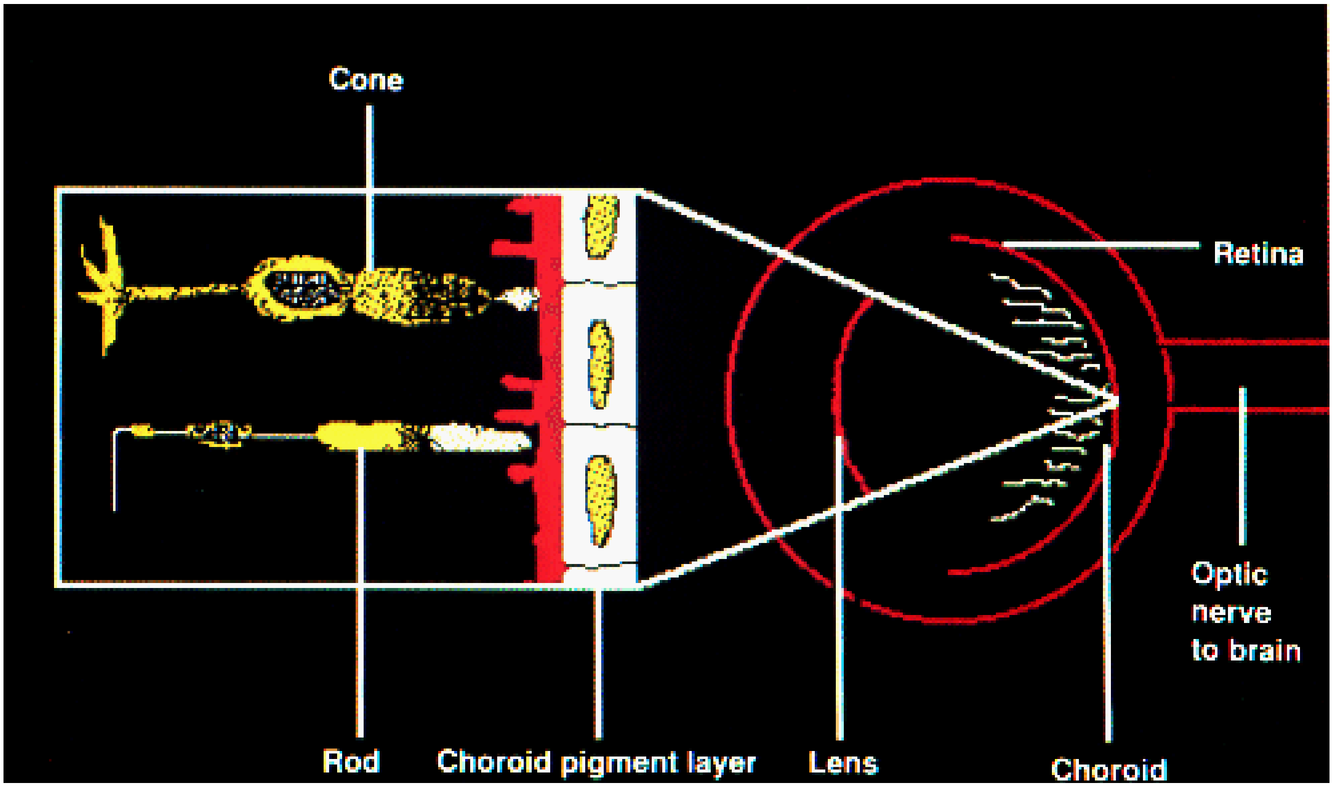


Figure 5: A schematic of rods and cones.

The *luminous-efficiency* function is the eye's response to light of constant luminance as the dominant wavelength is varied. Humans have peak sensitivity in the yellow-green light range around 550 nm.

2.2.2 Color matching

Imagine viewing a test color with the left eye and having 3 knobs to control the red, green, and blue components of a second color, seen with the right eye. Pretend you can turn the knobs turn in 1 nm increments. You are directed to turn the knobs until you agree the color seen with the right eye matches the test color seen with the left eye.

As the test color's dominant wavelength is varied, three color matching functions, r_λ, g_λ, and b_λ, are recorded to show the amount of red, green, and blue needed to match the test color. For some test colors, the values of r_λ, g_λ, or b_λ may be negative! Which is not possible (we can't suck up light). But we can move the appropriate red, green, or blue light to the test color side. Thus, it has been found that not all colors can be represented by positive RGB mixes.

2.2.3 The CIE color model

In 1931, the *Commission Internationale de l'Éclairage* (CIE) defined three primaries called **X**, **Y**, and **Z** to be used in color matching. These determine color matching functions x_λ, y_λ, and z_λ that remain positive for all matching of visible color. The **Y** primary's color matching function y_λ exactly matches the luminous-efficiency function. These color matching functions are tabulated at 1 nm intervals using color samples that subtend a 2° field of view on the retina. The amount of **X**, **Y**, and **Z** needed to match a color with spectral energy function $P(\lambda)$ are:

$$X = k\int P(\lambda)x_\lambda d\lambda, \quad Y = k\int P(\lambda)y_\lambda d\lambda, \quad Z = k\int P(\lambda)z_\lambda d\lambda.$$

For self-luminous objects, k is 680 lumens/watt. For reflecting objects

$$k = \frac{100}{\int P_w(\lambda)y_\lambda d\lambda},$$

where $P_w(\lambda)$ is the spectral energy function of "standard" white. The visible colors are contained in a cone-shaped region of **XYZ** space that extends from the origin into the positive octant.

Let color **C** be matched by

$$C = X\mathbf{X} + Y\mathbf{Y} + Z\mathbf{Z}.$$

Define *chromaticity values* by

$$x = \frac{X}{X+Y+Z}, \; y = \frac{Y}{X+Y+Z}, \; z = \frac{Z}{X+Y+Z}.$$

Given $(x,\, y,\, Y)$ we can recover $(X,\, Y,\, Z)$:

$$X = \frac{x}{y}Y, \quad Y = Y, \quad Z = \frac{1-x-y}{y}Y$$

Note $X+Y+Z$ can be thought of as the total amount of light energy The chromaticities depend only on dominant wavelength and excitation purity (they are independent of luminous energy)

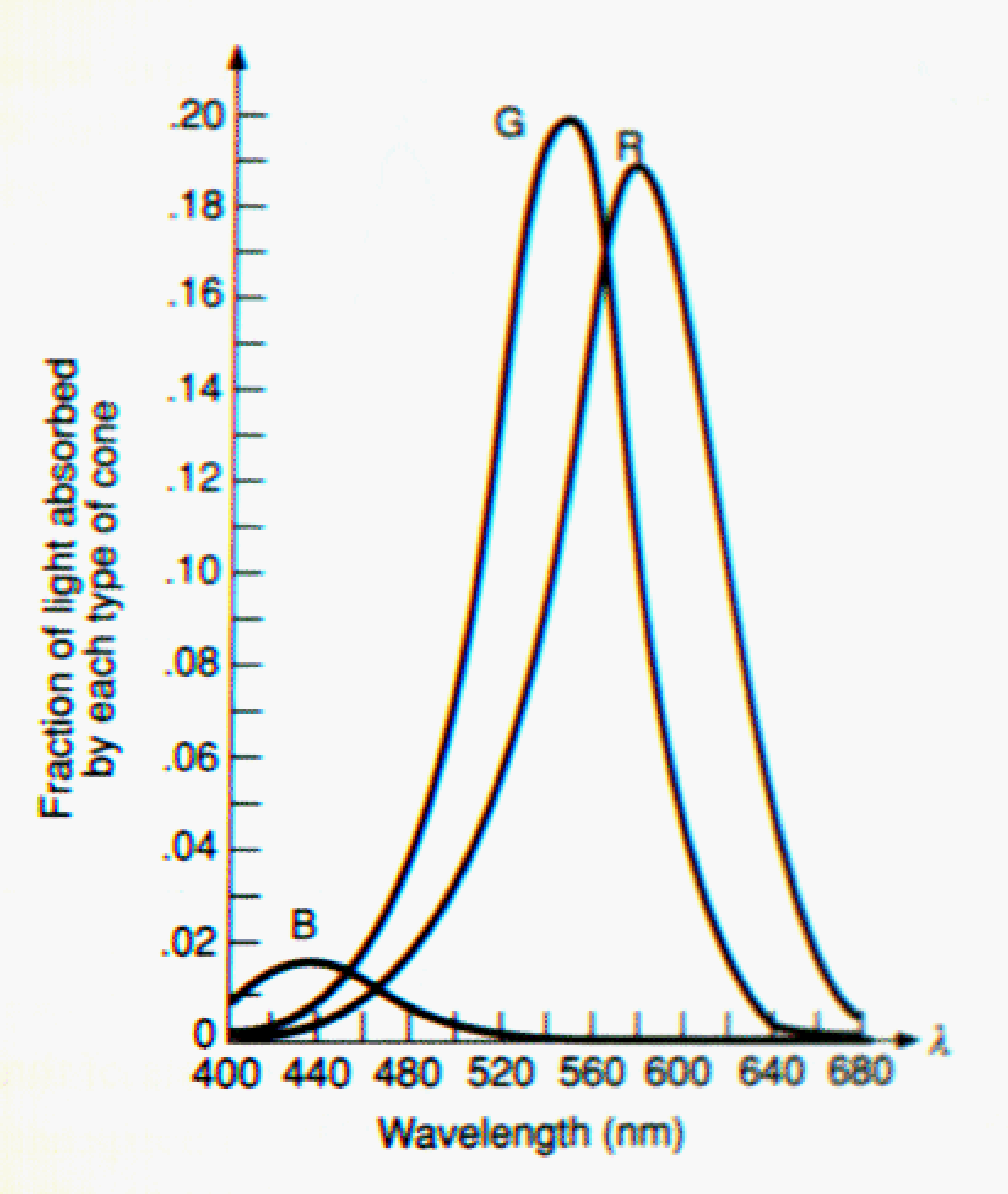


Figure 6: Sensitivity to spectral intensities.

The chromaticities are on the plane $\mathbf{X} + \mathbf{Y} + \mathbf{Z} = 1$ The orthographic projection of this plane onto the **XY** plane is called the *CIE chromaticity diagram*, see figure 7. The CIE chromaticity diagram is a plot of x and y for all visible colors. 100% spectrally pure colors are on the boundary of the CIE diagram. A standard white light, (approximately sunlight) called *illuminant C* is near where $x = y = z = 1/3$.

Complementary colors when mixed produce white. Complementary colors lie on opposite sides of a line through illuminant C. The dominant wavelength of a *nonspectral* color F is found by intersecting the line from the color through C with the "horseshoe" part of the CIE diagram. The excitation purity is the ratio of length CF to CG, where G is the closest point on the boundary along the line through C and F. A *color gamut* is the set of all colors that can be formed by mixing the colors — it is the convex hull of the colors in the CIE diagram. Color gamuts of different color devices can be compared using the chromaticity diagram. Color gamuts for monitors are triangular, film and print gamuts may be more complex shapes. The CIE LUV uniform color space was developed in 1976 to solve the problem that changes $\triangle C$ in color are not perceived to be equal.

2.2.4 The RGB color model

The red, green, and blue (RGB) color model is used for color CRT monitors. The model is represented by a unit cube, where red, green, and blue are at corners (1, 0, 0), (0, 1, 0), and (0, 0, 1), see figure 8. Black is at (0, 0, 0) with grays along the diagonal to white at (1, 1, 1).

The RGB *primaries* (red, green, and blue) are *additive*. That is new colors are formed by adding red, green, and blue. Cyan (0, 1, 1) is formed by adding green and blue; magenta (1, 0, 1) is formed by adding red and blue; yellow (1, 1, 0) is formed by adding red and green. Red and cyan are complementary; green and magenta are complementary; blue and yellow are complementary.

The color gamut of the RGB model is defined by the chromaticities of a CRT's phosphors. Let (x_r, y_r), (x_g, y_g), and (x_b, y_b) be the chromaticities for the vertices of a monitor's RGB color gamut. Let Y_r, Y_g, Y_b be the luminances of maximum-brightness red, green, and blue. Let $C_r = Y_r/y_r = X_r + Y_r + Z_r$, $C_g = Y_g/y_g = X_g + Y_g + Z_g$, $C_b = Y_b/x_b = X_b + Y_b + Z_b$. The transformation from RGB to CIE is given by

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_rC_r & x_gC_g & x_bC_b \\ y_rC_r & y_gC_g & y_bC_b \\ z_rC_r & z_gC_g & z_bC_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

where $z_r = 1 - x_r - y_r$ and similarly for z_g, z_b. If M_1 maps one monitor's RGB to CIE and M_2 maps another monitor's RGB to CIE, then $M_2^{-1}M_1$ maps monitor 1's RGB to monitor 2's RGB. Of course, some values may not be displayable on both monitors.

2.2.5 The CMY color model

Cyan, magenta, and yellow are used with hardcopy devices that deposit colored pigment onto paper. The CMY model describes this *subtractive* process for primaries cyan, magenta, and yellow. Cyan subtract (filters) red from reflected white light allowing only green and blue to be be seen. Magenta absorbs green; yellow absorbs blue. The CMY model uses a unit cube just as RGB does only with a relabeling of corners, cyan is at (1, 0, 0), magenta is at (0, 1, 0), and yellow is at (0, 0, 1).

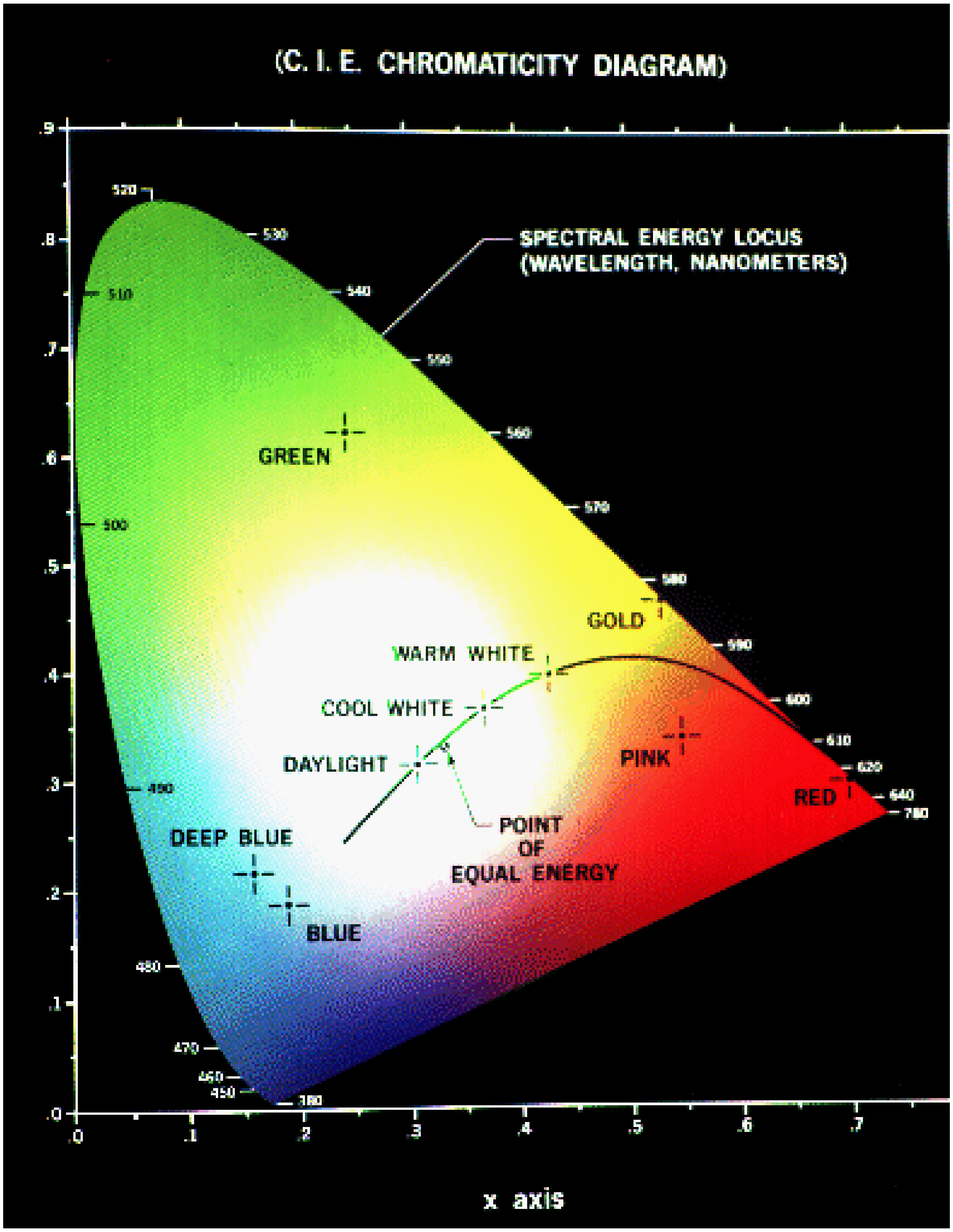


Figure 7: The CIE diagram.

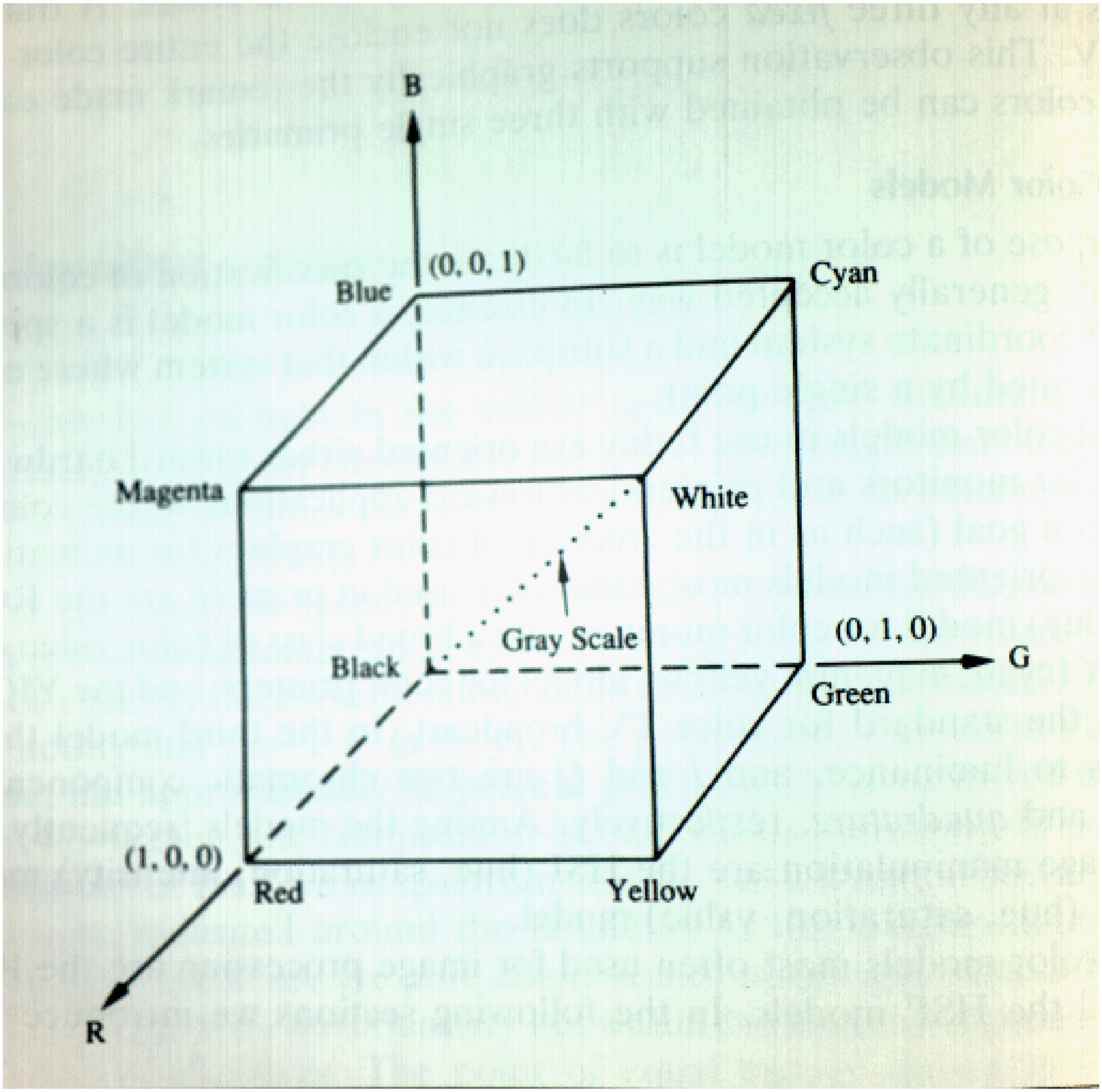


Figure 8: The RGB color cube.

The (ideal) transformations between CMY and RGB are given by

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}.$$

Of course, in practice it is not this simple to map correctly from CMY to RGB.

Four color printing uses the CMYK color model where K represents black. The (ideal) conversion from CMY to CMYK is

K = min(C, M, Y)
C = C - K
M = M - K
Y = Y - K

2.2.6 The HSV color model

The Hue, Saturation, and Value (HSV) color system is closer to the artist's model of color mixing. The model is defined on a *hexcone*, or six-sided pyramid. The top of the cone, given by $V = 1$, contains the brightest colors. The apex of the cone corresponds to black. Hue is measured by angles around the cone's vertical axis (Red = 0°, Green = 120°, Blue = 240°). Saturation ranges from 0 on the V axis to 1 on the cone's faces. The top of the hexcone is the projection of the RGB cube looking along the principal diagonal.

The conversion from RGB to HSV is non-linear. Value is determined by the largest RGB component, saturation is the relative range of RGB values, and the hue as the relative angular displacement from the largest RGB component. Here's some code that performs the transformation.

public hsvColor rgbTohsv (rgbColor color) hsvColor hsv = new hsvColor(); double max = maximum(color.red, color.green, color.blue); double min = minimum(color.red, color.green, color.blue); double delta = max - min; double hue = 0.0;

hsv.value = max; if (max != 0) hsv.saturation delta/max; else hsv.saturation = 0; if (hsv.saturation == 0) hsv.hue = UNDEFINED; else if color.red == max hue = (color.green - color.blue)/delta; else if color.green == max hue = 2+(color.blue-color.red)/delta; else hue = 4+(color.red - color.green)/delta; hsv.hue = hue * 60; if (hue ¡ 0) hsv.hue = hsv.hue + 360 ;

2.2.7 The HLS color model

The hue, lightness, and saturation (HLS) color model uses a double hexcone. White is at $L = 1$ and black is at $L = 0$. Hues are measured in angles from 0° to 360°. Saturation varies from 0 on the L axis to 1 on the face of the cone. Grays have $S = 0$. Maximally saturated color have $S = 1$, $L = 0.5$

Like the map from RGB to HSV, the map from RGB to HLS is non-linear.

public hlsColor rgbTohls (rgbColor color) hlsColor hls = new hlsColor(); double max = maximum(color.red, color.green, color.blue); double min = minimum(color.red, color.green, color.blue) double delta = max - min; double lightness = (max + min)/2.0;

if (max == min) s = 0.0; h = UNDEFINED; else if (lightness ¡ 0.5) s = (max - min)/(max + min); else s = (max - min)/(2-max + min); if (color.red == max) h = (color.green - color.blue)/delta; else if (color.green == max) h = 2+(b-r)/delta; else h = 4+(r-g)/delta; h= h * 60; if h ¡ 0 h = h+360;

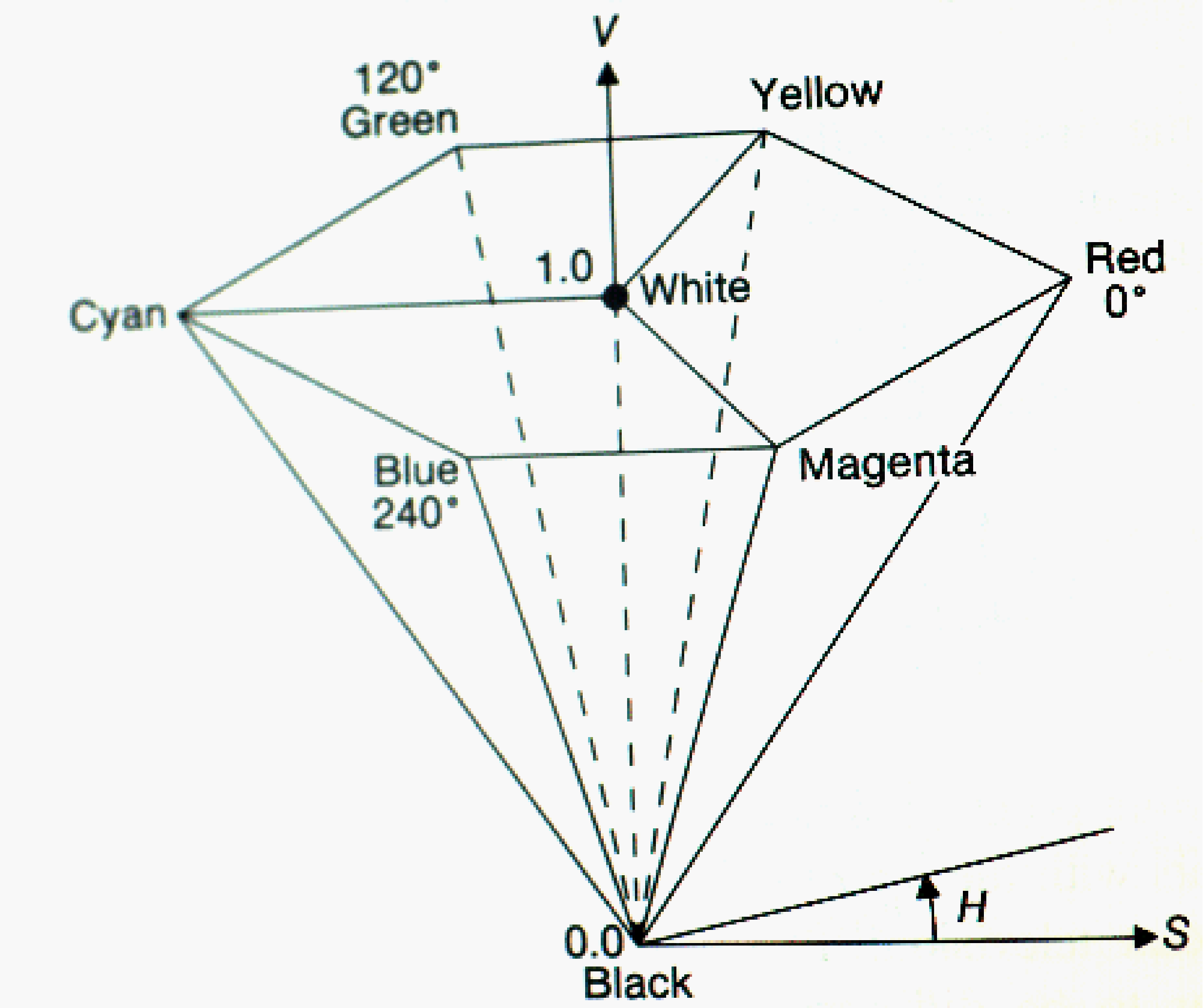


Figure 9: The HSV color model.

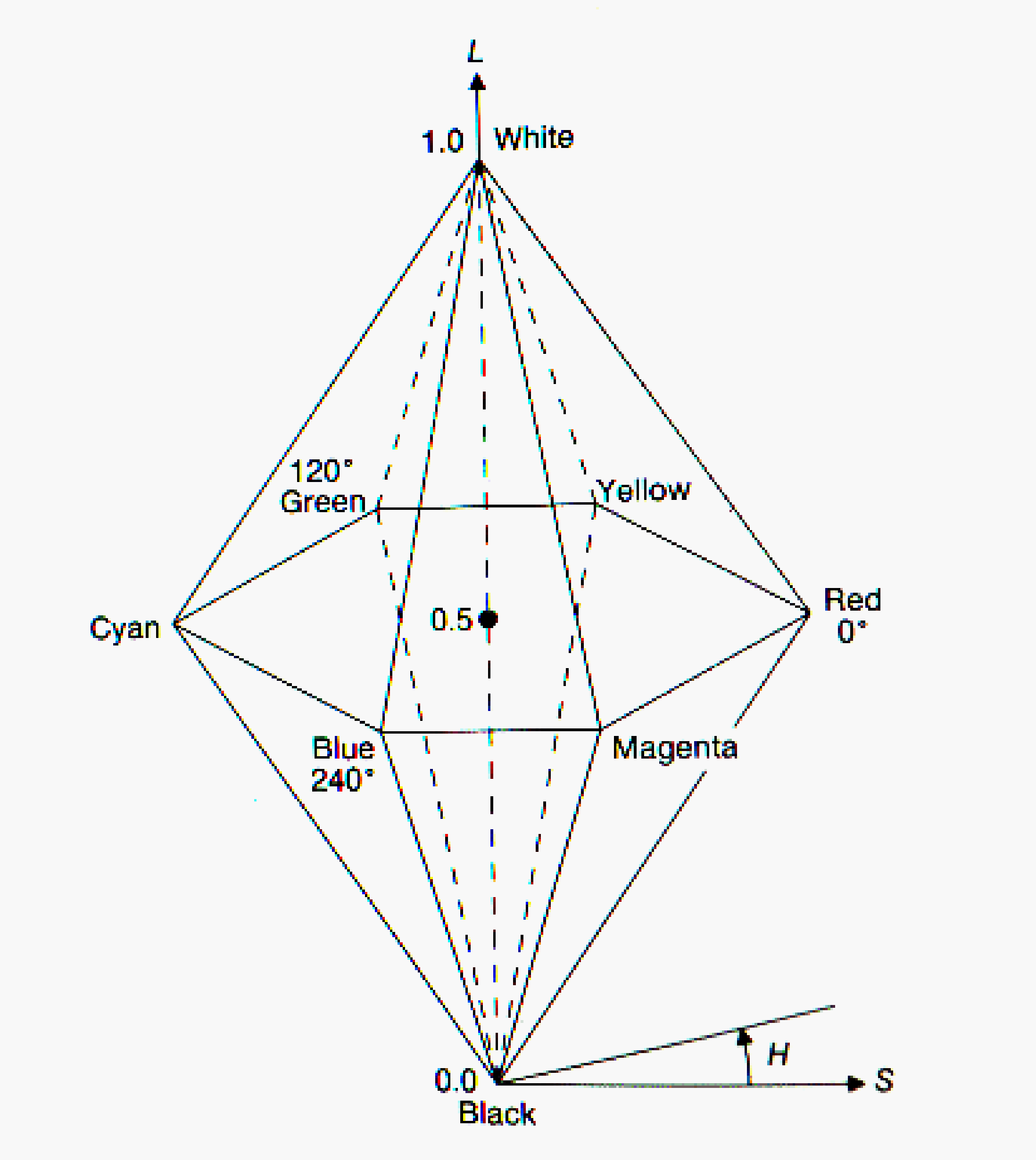


Figure 10: The HLS color model.

2.2.8 The CNS color model and Munsell colors

The *color naming system* (CNS) is based on natural language color categories.

- Lightness is chosen from *very dark*, *dark*, *medium*, *light* , or *very light*
- Saturation can be *grayish*, *moderate*, *strong*, or *vivid*
- Thirty-one hues are formed from seven generic hues: *red*, *orange*, *brown*, *yellow*, *green*, *blue*, and *purple*
- Halfway hues are denoted by hyphenation, eg, green-blue
- Quarterway hues with an *ish* suffix, eg yellowish-green

The Munsell color-order system is a set of published standard colors organized in a space of hue, value, and chroma (saturation) Each color in the Munsell system is named and ordered to have an equally perceived "distance" in color space from its neighbors.

2.2.9 Interpolating in color space

Color interpolation needed for (1) Gouraud shading, (2) antialiasing, and (3) blending images (fade-in, fade-out). Results of interpolation depend on color model being use. If map from one color model to another is linear, then linear interpolation in both models will reproduce identical colors, for example, interpolation in spaces RGB, CMY, CIE, and YIQ produce equivalent colors. However, the map from RGB to HSV or HLS is non-linear and color interpolation is not valid between spaces. For example, Consider $(1 - t)(1, 0, 0) + t(0, 1, 0)$ which linearly interpolates from red to green in RGB space. At $t = 0.5$ the color $(0.5, 0.5, 0)$ maps to $(60°, 1, 0.5)$ in HSV. Now consider $(1-t)(0°, 1, 1)+t(120°, 1, 1)$ which linearly interpolates from red to green in HSV space. At $t = 0.5$ the color $(60°, 1, 1)$!

For Gouraud shading any model can be use because the two interpolants are usually close together. For antialiasing and fade-in, fade-out then RGB is appropriate.

2.2.10 Reproducing color

Spatial integration of RGB triad on color monitors and 4-color (CMY+Black) printing. Dithering – use 2×2 (or larger grids of pixels)

- Extend color range, but lose resolution
- If 3-bits/pixel (1 for red, green and blue), then there are 5 different reds, greens and blues or 125 color combinations.

Xerography, ink-jets and thermal color copiers actually mix subtractive color pigments.

- Given a color image with n bits/pixel to be displayed on a device with $m < n$ bits/pixel
  - Which 2^m colors should be displayed?
  - What map from the 2^n colors in the image to the 2^m colors should be used?

  - Solution 1: Use a fixed set of display colors and a fixed mapping (eg. $m = 8$, use 3 bit for red and green, 2 bits for blue)
  - Solution 2: *popularity algorithm* — create histogram of the image's color and use the 2^m most frequent colors.
  - Solution 3: *median-cut algorithm* — recursively fit a box around the colors used in the image, splitting the box along its longer dimension at the median until 2^m boxes have been created. Use the centroid of the box as the display color for all image colors in the box.

- Accurate color reproduction is difficult.

2.2.11 Guidelines for using color

Here's a list of guidelines on the use of color in computer graphics.

1. Select colors by some method, e.g. traversing a smooth curve or restricting to a plane in color space.
2. Avoid the simultaneous display of highly saturated, spectrally extreme colors.
3. Pure blue should be avoided for text, thin lines, and small shapes.
4. Avoid adjacent colors that differ only in the amount of blue.
5. Older operators need higher brightness levels to distinguish colors.
6. Colors change in appearance as the ambient light level changes.
7. The magnitude of a detectable change in color varies across the spectrum.
8. It is difficult to focus upon edges created by color alone.
9. Avoid red and green in the periphery of large-scale displays.
10. Opponent colors go well together.
11. For color-deficient observers, avoid single-color distinctions.
12. Random selection of different hues is usually garish.
13. If a display contains only a few colors, the complement of one color should be used as the background.
14. A neutral (gray) should be used for the background for an image with many colors.
15. If two adjoining colors are not harmonious, place thin black border between them.

Charles Poynton has an extensive collection of references on color. You may which to peruse them at http://www.inforamp.net/%7Epoynton/notes/links/color-links.html.

3 Illumination Models

An *illumination model (equation)* expresses the components of light reflected from or transmitted (refracted) through a surface. There are three basic light components: *ambient*, *diffuse*, and *specular*. We will develop *local* illumination models that contain some or all of these components. The models are local in that they do not consider light from objects in the scene. Only light sources generate light. Light reflected from other objects does has no effect on other objects. Ray tracing and radiosity models provide these global lighting effects.

Only a crude approximation to ambient light will be used to represent light from *environment* and its effect on the light reflected from an object. Light can be diffusely or specularly reflected and diffusely or specularly refracted.

The illumination equation must be evaluated in view space since perspective mapping destroys the geometry of surface normals, view vectors, light source vectors.

3.1 Ambient Reflected Light

- *Ambient* light component — non-directional light source that is the product of multiple reflections from the surrounding environment
  - Assume the intensity I_a of ambient light is constant for all objects
  - The *ambient-reflection coefficient* k_a, which ranges from 0 to 1, determines the amount of ambient light reflected by the object's surface
  - The ambient-reflection coefficient is a *material property*
  - The ambient illumination equation is

    $$I = k_a I_a$$

    where I is the intensity of reflected light from a surface with ambient-reflection coefficient k_a in an environment with ambient intensity I_a

3.2 Diffuse light

Diffuse light is reflected (or transmitted) from a point with equal intensity in all directions. Diffusely reflected light is typical for dull, matte surfaces such a paper or a flat wall paint. Diffusely refracted light is typical for frosted glass.

Diffuse reflection is modeled by the *Lambert's laws*, which basically states that brightness depends only on the angle θ between the light source direction $\vec{L}$ and the surface normal $\vec{N}$.

Lambert's first law: The illuminance on a surface illuminated by light falling on it perpendicularly from a point source is proportional to the inverse square of the distance between the surface and the source.

Lambert's second law: If the rays meet the surface at an angle, then the illuminance is proportional to the cosine of the angle with the normal.

Lambert's third law: The luminous intensity of light decreases exponentially with distance as it travels through an absorbing medium.

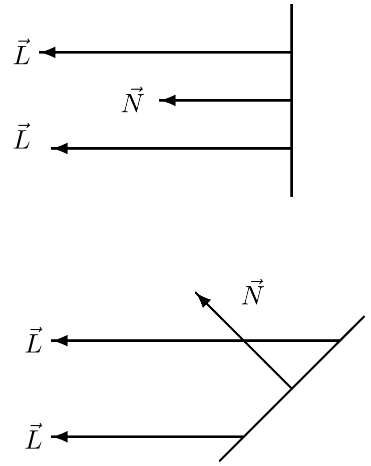


Figure 11: A diagram illustrating Lambert's Laws.

The light beam covers an area whose size is inversely proportional to the cosine of θ and the amount of reflected light seen by the viewer is *independent of the viewer's position* and proportional to $\cos\theta$.

Diffuse illumination model. What all this means is that the diffuse illumination equation we'll use is

$$I = k_d I_p \cos\theta,$$

where

1. θ is the angle between the surface normal $\vec{N}$ and the light vector $\vec{L}$;
2. I_p is the intensity of a point light source; and
3. k_d is the *diffuse-reflection coefficient* or *diffuse reflectivity* of the surface which varies between 0 and 1 and depends on the surface material.

Assuming that $\vec{N}$ and $\vec{L}$ are unit length vectors, we can write the cosine as a simple dot product and the diffuse illumination equation as

$$I = k_d I_p(\vec{N} \cdot \vec{L}).$$

This equation must be evaluated in view coordinates since a perspective transformation will modify θ and the inner product. Often, we pretend the point light source is sufficiently far away so that $\vec{L}$ can be considered constant for all points on the surface. Thus, the diffuse illumination equation only needs to be evaluated once for the entire surface, provided of course the surface is flat so that $\vec{N}$ does not change either.

Intensity attenuation. We should also take into account the distance of objects from the light source. This will allow us to distinguish two identical, parallel surfaces at different distances. The intensity from the more distance surface must be attenuated (lessened). The energy from a point light source obeys an inverse square law, that is our basic diffuse illumination model should be modified to

$$I = \frac{k_d I_p (\vec{N} \cdot \vec{L})}{d^2},$$

where d is the distance from the point light source to the surface. In practice, a more general model usually gives better results. For example, Foley and Van Dam [1] suggest using

$$\max\left(\frac{1}{c_1 + c_2 d + c_3 d^2},\, 1\right)$$

as an attenuation factor for some choice of "tuning" parameters c_1, c_2, and c_3. So now our diffuse model is

$$I = \frac{k_d I_p (\vec{N} \cdot \vec{L})}{\min\left(c_1 + c_2 d + c_3 d^2,\, 1\right)}.$$

A more simple model, such as,

$$I = \frac{k_d I_p (\vec{N} \cdot \vec{L})}{c_1 + c_2 d},$$

is also often used.

3.3 Specular Light

The *specular* component of reflection (or refracted) light accounts for highlights caused by light reflecting (or refracting) primarily in one direction. Specular reflection is mirror-like; it gives rise to shiny spots on surfaces. The amount of specularly reflected light seen by a viewer depends on the angle θ between $\vec{L}$ and $\vec{N}$ and the angle ϕ between the viewer $\vec{V}$ and the reflected ray $\vec{R}$. Specularly reflected light, unlike diffusely reflected light, in *view dependent.* In drawing the specular component we often refer to the "specular bump" which shows that most light is reflected in a particular direction.

Bui-Thong Phong developed a popular approximation to the specular component of light. Phong's model is

$$I = W(\theta) I_p \cos^n(\phi),$$

where $W(\theta)$ is the fraction striking the surface that is specularly reflected, and n is the *Phong specular-reflection exponent.* $W(\theta)$ is often set to a constant, call it k_s, and refer to it as the *specular-reflection coefficient* or *specular reflectivity.* between 1 and several The Phong exponent n varies from 1 to several hundred. A setting $n = 1$ gives broad gentle falloff to the highlight, while large settings of n give focused highlight.

3.4 The complete illumination model

Light is additive. To model reflected light we simply add the ambient, diffuse, and specular terms. That is, our basic illumination model is

$$I_\lambda = k_a I_{a\lambda} O_{d\lambda} + f_{att} I_{p\lambda} \left[k_d O_{d\lambda} (\vec{N} \cdot \vec{L}) + k_s O_{s\lambda} (\vec{R} \cdot \vec{V})^n \right]$$

where $O_{s\lambda}$ is the object specular color component

3.5 Calculating the normal vector $\vec{N}$

Since we are only considering polygonal objects it is simple to calculate their normal vectors. Differentiation would be required for more general surfaces.

A planar polygon lies in some plane with an equation

$$ax + by + cz + d = 0,$$

for some constants a, b, c, and d. A vector perpendicular to the plane is

$$\langle a,\, b,\, c\rangle.$$

Since we want our normal vector to have unit length, the normal vector is

$$\vec{N} = \langle \frac{a}{\sqrt{a^2+b^2+c^2}}, \frac{b}{\sqrt{a^2+b^2+c^2}}, \frac{c}{\sqrt{a^2+b^2+c^2}} \rangle.$$

We're typically given only the coordinates of the polygon's vertices, call them $v_0, \ldots, v_{n-1}$. How do we calculate the normal vector from these? We'll, given any two vectors in the plane, say $v_1 - v_0$ and $v_2 - v_0$, their cross product $(v_1 - v_0) \times (v_2 - v_0$ will be perpendicular (orthogonal) to the plane. Scaling this cross-product to unit length produces the polygon's normal vector.

But there is another potential problem: A polygon has two sides, which we will call "front" and "back." Most often we can see the front side, but not the back side. There are two normals a *front normal* and *back normal*.

Newell's technique for computing the *outward pointing surface normal* $\vec{N} = \langle N_x,\, N_y,\, N_z\rangle$

$$N_x = \sum_{k=1}^{n} (y_k - y_{k+1})(z_k + z_{k+1})$$

$$N_y = \sum_{k=1}^{n} (z_k - z_{k+1})(x_k + x_{k+1})$$

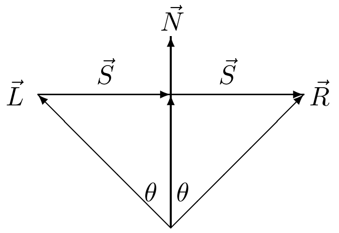
$$N_z = \sum_{k=1}^{n} (x_k - x_{k+1})(y_k + y_{k+1})$$

where $P_k = (x_k,\, y_k,\, z_k)$, $k = 1, \ldots, n$ are the vertices of the polygon and $P_{n+1} = P_1$

3.6 Calculating the reflection vector $\vec{R}$

- The reflection vector $\vec{R}$ is the mirror image of the unit light vector $\vec{L}$ about the unit surface normal $\vec{N}$
- The projection of $\vec{L}$ onto $\vec{N}$ is $\vec{N}\cos\theta$
- Let $\vec{S} = \vec{N}\cos\theta - \vec{L}$

- Note that $\vec{R} = \vec{N}\cos\theta + \vec{S}$
- Thus $\vec{R} = 2\vec{N}\cos\theta - \vec{L}$
- Or $\vec{R} = 2\vec{N}(\vec{N}\cdot\vec{L}) - \vec{L}$
- And $\cos\alpha = (\vec{R}\cdot\vec{V}) = 2(\vec{N}\cdot\vec{V})(\vec{N}\cdot\vec{L}) - (\vec{L}\cdot\vec{V})$



The Halfway Vector

- Blinn proposed an alternative to Phong's model that uses the *halfway vector*

$$\vec{H} = \frac{\vec{L}+\vec{V}}{\mid \vec{L}+\vec{V} \mid}$$

  which is halfway between the light source and viewer
- The halfway vector is the direction of maximum highlights
- If the surface is oriented so that the normal is aligned along $\vec{H}$, the viewer sees the brightest specular highlights
- The specular term is $(\vec{N}\cdot\vec{H})^n$
- The illumination equation is

$$I_\lambda = k_a I_{a\lambda} O_{d\lambda} + f_{att} I_{p\lambda}\left[k_d O_{d\lambda}(\vec{N}\cdot\vec{L}) + k_s O_{s\lambda}(\vec{N}\cdot\vec{H})^n\right]$$

3.7 Chromatic Light and Multiple Sources

Until now, we've only considered the intensity of reflected light.

- Colored lights and surfaces can be modeled by sampling the intensity at discrete wavelengths

$$I_\lambda = k_a I_{a\lambda} O_{d\lambda} + f_{att} k_d I_{p\lambda} O_{d\lambda}(\vec{N}\cdot\vec{L})$$

- Here $O_{d\lambda}$ defines the object's *diffuse color component* at wavelength λ
- Often only three, red, green, and blue, components are sampled, which leads to color aliasing, but is acceptable for simple renderings

- That is, a red intensity I_{red}, a green intensity I_{green}, and a blue intensity, I_{blue} are used to define the color in the RGB color system
- In theory, the intensity should in integrated over the visible spectrum
- Multiple light sources can be modeled by summing the intensity reflected by the object for each source

$$I_\lambda = k_a I_{a\lambda} O_{d\lambda} + \sum_{j=1}^{n} f_{att_j} k_d I_{p\lambda j} O_{d\lambda} (\vec{N} \cdot \vec{L_j})$$

3.8 Physically Based Illumination Models

- The illumination model (equation) developed so far is based on common-sense and works well in many situation
- Physical laws can be applied to develop better, but more expensive models
- Several definitions needed
  - *Flux* is the rate at which light energy is emitted (watts)
  - A *solid angle* is the angle at the apex of a cone and can be defined in terms of the area on a hemisphere intercepted by a cone with vertex at the hemisphere's center (just as radians are defined in terms of arc length)
  - A *steradian* is the solid angle of a cone that intercepts an area equal to the square of hemisphere's radius r (there are 2π steradians (sr) in a hemisphere)
  - *Radiant intensity* is the flux radiated into a unit solid angle in a particular direction (watts/sr)
  - *Intensity* is the radiant intensity for a point light source
  - *Foreshortened surface area* is computed by multiplying the surface area by $\cos\theta_r$, where θ_r is the angle of the radiated light relative to the surface normal
  - *Radiance* is the radiant intensity per unit foreshortened surface area $(W/(sr \cdot m^2)$
  - *Irradiance* is the flux per unit surface area (W/m^2)
  - The irradiance of incident light is

    $$E_i = I_i (\vec{N} \cdot \vec{L}) d\omega_i$$

    where I_i is the incident light's radiance
  - The *bidirectional reflectivity* ρ is the ratio of reflected radiance (intensity) in one direction to the incident irradiance (flux density) responsible for it from another direction

    $$\rho = \frac{I_r}{E_i}$$

  - The reflected radiance is

    $$I_r = \rho I_i (\vec{N} \cdot \vec{L}) d\omega_i$$

- Bidirectional reflectivity is composed of diffuse and specular components

$$\rho = k_d \rho_d + k_s \rho_s, \quad k_d + k_s = 1$$

- The illumination equation for the reflected intensity is from n light sources is

$$I_r = \rho_a I_a + \sum_{j=1}^{n} I_{i_j} (\vec{N} \cdot \vec{L}_j) d\omega_{i_j} (k_d \rho_d + k_s \rho_s)$$

3.8.1 The Torrance-Sparrow illumination model

- Developed by physicists and introduced by Blinn into computer graphics to better model the specular reflectivity
- Surface is a collection of planar *microfacets*, each a perfectly smooth reflector
- The geometry and distribution of the microfacets and the direction of the light determine the intensity and direction of specular reflection
- The Torrance-Sparrow model assumes the specular component of bidirectional reflectivity is given by

$$\rho_s = \frac{F_\lambda D G}{\pi (\vec{N} \cdot \vec{V})(\vec{N} \cdot L)}$$

  where

  - D is the distribution function of the microfacet orientation
  - G is the *geometrical attenuation factor* (shadowing and masking of microfacets on each other)
  - F_λ is the Fresnel term which relates the incident light to reflected light for a smooth surface
  - The $\vec{N} \cdot \vec{V}$ term make the equation proportional to the surface area seen by the viewer in a unit of foreshortened area
  - The $\vec{N} \cdot \vec{L}$ term make the equation proportional to the surface area the light sees in a unit of foreshortened area
  - The π accounts for surface roughness

The microfacet distribution

- The microfacet distribution D determines the roughness of the surface
- Torrance and Sparrow assume the microfacet distribution is Gaussian

$$D = c_1 e^{-(\beta/m)^2}$$

  where c_1 is an arbitrary constant, m is the root mean square slope of the microfacets, and β is the angle between $\vec{N}$ and $\vec{H}$

- Cook and Torrance use the more theoretically correct Beckman distribution

$$D = \frac{1}{m^2 \cos^4 \beta} e^{-(\tan \beta / m)^2}$$

- When m is small, the microfacets vary slightly from the surface normal and the reflection is highly directional
- When m is large, the microfacets slopes are steep and the rough surface spreads out the light

The geometric attenuation factor

- The geometric attenuation factor $0 \leq G \leq 1$ takes into account that some microfacets will lie in the shadow of others or light reflected from them will strike other microfacets
- If the reflected light from the microfacet is not blocked from the view and not in shadow, then $G = 1$
- Let l denote the area of a microfacet and m the area whose reflected light is blocked or is in shadow

$$G_{b|s} = 1 - \frac{m}{l}$$

- For light blocked from the viewer,

$$G_b = \frac{2(\vec{N} \cdot \vec{H})(\vec{N} \cdot \vec{V})}{(\vec{V} \cdot \vec{H})}$$

- For light in shadow

$$G_s = \frac{2(\vec{N} \cdot \vec{H})(\vec{N} \cdot \vec{L})}{(\vec{V} \cdot \vec{H})}$$

- G is the minimum of these 3 values

$$G = \min \left\{ 1, \frac{2(\vec{N} \cdot \vec{H})(\vec{N} \cdot \vec{V})}{(\vec{V} \cdot \vec{H})}, \frac{2(\vec{N} \cdot \vec{H})(\vec{N} \cdot \vec{L})}{(\vec{V} \cdot \vec{H})} \right\}$$

The Fresnel term

- The Fresnel equation for unpolarized light specifies the ratio of reflected light from a dielectric (non-conducting) surfaces as

$$F_\lambda = \frac{1}{2} \left(\frac{\sin^2(\theta_i - \theta_t)}{\sin^2(\theta_i + \theta_t)} + \tan^2(\theta_i - \theta_t) \tan^2(\theta_i - \theta_t) \right)$$

where $\theta_i = \cos^{-1}(\vec{L} \cdot \vec{H})$ and θ_t is the angle of refraction

$$\sin \theta_t = (\eta_{i\lambda} / \eta_{t\lambda}) \sin \theta_i$$

where $\eta_{i\lambda}$ and $\eta_{t\lambda}$ are the indices of refraction for the two media

- The Fresnel term can be written as

  $$F_\lambda = \frac{1}{2}\frac{(g-c)^2}{(g+c)^2}\left(1+\frac{[c(g+c)-1]^2}{[c(g-c)-1]^2}\right)$$

  where $c = \vec{L}\cdot\vec{H}$, $g^2 = \eta_\lambda^2 + c^2 - 1$, and $\eta_\lambda = \eta_{i\lambda}/\eta_{t\lambda}$ is the index of refraction
- For conducting medium, the index of refraction is expressed as a complex number
- The color of the specular reflection the surface material, the incident light's wavelength and its angle of incidence
- When $\vec{H} = \vec{L} = \vec{V}$, $\theta_i = 0$, so $c = 1$, $g = \eta_\lambda$ and

  $$F_{\lambda_0} = \left(\frac{\eta_\lambda - 1}{\eta_\lambda + 1}\right)^2$$

- When $\vec{H} \perp \vec{L}$, $\theta_i = \pi/2$, so $c = 0$ and

  $$F_{\lambda_{\pi/2}} = 1$$

  , (the color of the material is the color of the light)
- Specular reflectance ρ_s depends on η_λ for all angles except $\pi/2$
- η_λ can be derived from F_{λ_0},

  $$\eta_\lambda = \frac{1+\sqrt{F_{\lambda_0}}}{1-\sqrt{F_{\lambda_0}}}$$

4 Shading Methods

- *Flat shading*, also called *constant shading* or *faceted shading*, is the most simple approach
  - Apply the illumination equation once for each polygon
  - Approach is valid if:
    - The light source is at infinity, so $\vec{N}\cdot\vec{L}$ is constant for a face
    - The viewer is at infinity, so $\vec{N}\cdot\vec{L}$ is constant for a face
    - The polygon is the actual surface being modeled (not an approximation to a curved surface)
- By contrast, interpolated shading computes a separate intensity for each point on the polygon
  - Assume a curved surface is approximated by a polygonal mesh
  - Interpolated can be used to smooth the facets by continuously changing the intensity across edges
  - *Mach bands* appear where intensity has a change in magnitude or slope (dark facets are perceived to be darker and light facets are perceived to be lighter

4.1 Gouraud Shading

- Gouraud shading is an *intensity-interpolation* method
- The intensity is calculated at each vertex of a polygon and linearly interpolated across the polygon's edges and face
- This allows the use of polygonal models to represent smooth surfaces such as cylinders and spheres
- We approximate the normals at the polygon vertices by adding all normals for each surface that meets at the vertex
- Consider a truncated pyramid with vertices

$$\begin{aligned}
V_1 &= (-1,-1,1)\\
V_2 &= (1,-1,1)\\
V_3 &= (1,1,1)\\
V_4 &= (-1,1,1)\\
V_5 &= (-2,-2,0)\\
V_6 &= (2,-2,0)\\
V_7 &= (2,2,0)\\
V_8 &= (-2,2,0)
\end{aligned}$$

- And plane equations surrounding vertex V_1:

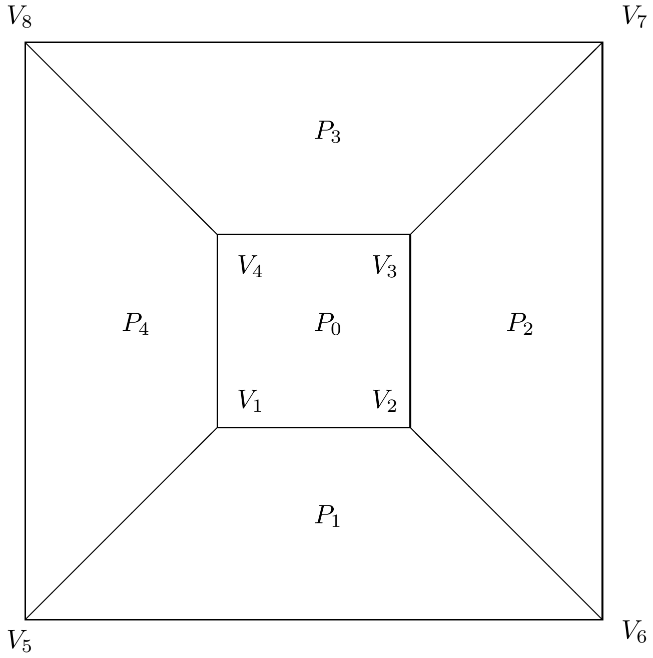
$$\begin{array}{rrcl}
P_0: & z-1 & = & 0\\
P_1: & -y+z-2 & = & 0\\
P_4: & -x\quad +z-2 & = & 0
\end{array}$$

- The normal direction at V_1 is then approximately

$$N = ([0+0-1],[0-1+0],[1+1+1]) = (-1,-1,3)$$

or normalized

$$\hat{N} = (-1,-1,3)/\|N\| = (-1,-1,3)/\sqrt{11} \approx (-0.3,-0.3,0.9)$$



- Given an approximate normal at each vertex of the polygon, we shade the image a scan line at a time.
- The intensity at Q is approximated as a linear combination of the intensities at A and B,

  $$I_Q = (1-u)I_A + uI_B \qquad 0 \le u \le 1$$

  where

  $$u = \|AQ\|/\|AB\|$$

- Similarly at R, we approximate the intensity by

  $$I_R = (1-w)I_B + wI_C \qquad 0 \le w \le 1$$

  where

  $$w = \|BR\|/\|BC\|$$

- To compute the intensity at P between Q and R we interpolate the intensities at Q and R,

  $$I_P = (1-t)I_Q + tI_R \qquad 0 \le t \le 1$$

  where

  $$t = \|QP\|/\|QR\|$$

- The intensity calculation along a scan line can be done incrementally.
- For two adjacent pixels P_1 and P_2 we have

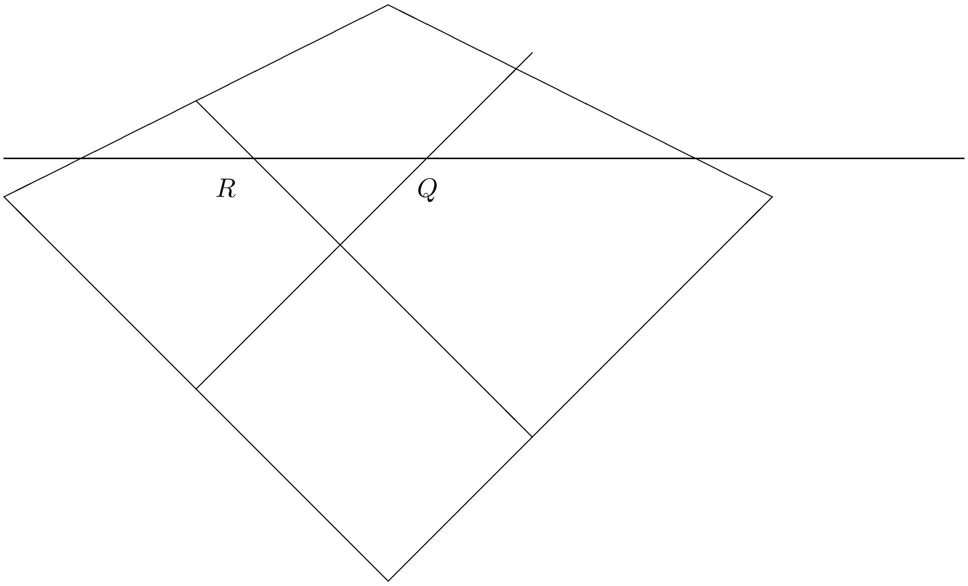
  $$I_{P_2} = (1-t_2)I_Q + t_2 I_R$$

and

$$I_{P_1} = (1 - t_1)I_Q + t_1 I_R$$

Subtracting yields

$$I_{P_2} = I_{P_1} + (I_R - I_Q)(t_2 - t_1) = I_{P_1} + \triangle I \triangle t$$



- Gouraud shading gives a improvement over constant shading
- Silhouette edges belie the smooth appearance
- Mach bands can still appear
- Normals along faces can point in radically different direction, but the average may all point in the same direction giving a flat appearance to a non flat object.
- The orientation of the polygon affects its shading
- Specular effects are not represented well

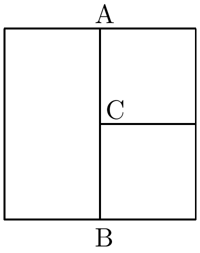
4.2 Phong Shading

- Phong shading is normal-vector interpolation shading
- The normals at vertices are computed as in Gouraud shading

- The normals are interpolated across polygon edges and between edges on a scan line
- The interpolation can be done incrementally (as the intensities were for Gouraud shading
- At each pixel, the normal is made to be unit vector and inversed mapped into world coordinates
- The illumination equation is evaluated to determine the intensity at the pixel
- Phong shading is better than Gouraud when specular components appear in the illumination model
  - Gouraud shading may miss highlight altogether if they don't appear at vertices
  - Gouraud shading spread highlights at a vertex across the surface
  - Mach bands are reduced in most cases
- Phong shading is expensive

4.3 Problems with interpolated shading

- Polygonal silhouette edges still appear, for example, in the approximation of a sphere
- Perspective distortion occurs because interpolation is not performed in world coordinates
- The interpolated values depend on the object's orientation
- Shading discontinuities occur when two adjacent polygons fail to share a vertex that lies along their common edge



- Vertex normals may not represent the surface's geometry, for example, averaged normals may all point in the same direction

4.4 Transparency

- Not all surfaces are opaque — some transmit light, e.g. glass, water
- Light ray is "bent" as it passes from one medium to another

- Snell's law states that the refracted (transmitted) ray lies in the same plane as the incident ray and governed by the relationship

  $$\eta_i \sin\theta_i = \eta_t \sin\theta_t$$

  where η_i and η_t are the indices of refraction of the two media, and θ_i is the angle between the incident ray and the surface normal and θ_t is the angle between the transmitted ray and the surface normal

- Light can be transmitted specularly (transparent) or diffusely (translucent)

- *Interpolated transparency* computes the intensity at a pixel where transparent surface 1 covers surface 2 by

  $$I_\lambda = (1 - k_{t1})I_{\lambda 1} + k_{t1} I_{\lambda 2}$$

  where $I_{\lambda 1}$ and $I_{\lambda 2}$ are the intensities of the two surfaces and k_{t1} is the transmission coefficient of surface 1 ($1 - k_{t1}$ is the surface's *opacity*)

- *Filtered transparency* is modeled by

  $$I_\lambda = I_{\lambda 1} + k_{t1} O_{t\lambda} I_{\lambda 2}$$

  where $O_{t\lambda}$ is polygon 1's *transparency color*

- Some visible surface algorithms can implement these transparency equations easily, other visible surface algorithm make it difficult to implement transparency

Calculating the Refraction Vector

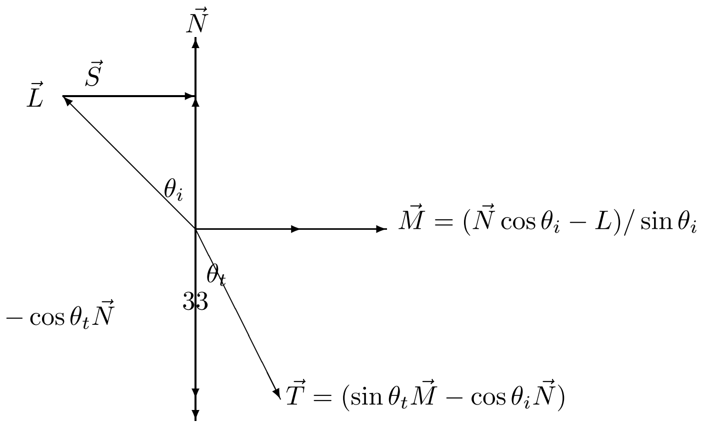
- The (unit) refraction vector is calculated as

  $$\vec{T} = \sin\theta_t \vec{M} - \cos\theta_t \vec{N}$$

  where $\vec{M}$ is a unit vector perpendicular to $\vec{N}$ in the plane of the incident light ray $\vec{L}$ and $\vec{N}$

- Note that $\vec{M} = \vec{S}/\sin\theta_i = (\vec{N}\cos\theta_i - L)/\sin\theta_i$, so

  $$\vec{T} = \frac{\sin\theta_t}{\sin\theta_i}(\vec{N}\cos\theta_i - \vec{L}) - \cos\theta_t \vec{N}$$


$\vec{N}$
$\vec{S}$
$\vec{L}$
θ_i
$\vec{M} = (\vec{N}\cos\theta_i - L)/\sin\theta_i$
θ_t
$-\cos\theta_t \vec{N}$
$\vec{T} = (\sin\theta_t \vec{M} - \cos\theta_i \vec{N})$

- The index of refraction is
$$\eta_\lambda = \eta_{i\lambda}/\eta_{t\lambda} = \sin\theta_t/\sin\theta_i$$
- Thus
$$\vec{T} = (\eta_\lambda \cos\theta_i - \cos\theta_t)\vec{N} - \eta_\lambda \vec{L}$$
- Note that
$$\cos\theta_i = \vec{N} \cdot \vec{L}$$
and
$$\cos\theta_t = \sqrt{1 - \sin^2\theta_t} = \sqrt{1 - \eta_\lambda^2 \sin^2\theta_i} = \sqrt{1 - \eta_\lambda^2(1 - (\vec{N} \cdot \vec{L})^2)}$$
- Thus
$$\vec{T} = \left(\eta_\lambda(\vec{N} \cdot \vec{L}) - \sqrt{1 - \eta_\lambda^2(1 - (\vec{N} \cdot \vec{L})^2)}\right)\vec{N} - \eta_\lambda \vec{L}$$

5 Conclusions

References

[1] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics Principles and Practice*, Addison–Wesley, second in c ed., 1995. 0-201-84840-6.