3

Color and Lighting

Chapter Objectives:

- Introduce RGB color in the hardware, eye characteristics, and gamma correction
- Understand color interpolation and smooth shading in OpenGL
- Set up OpenGL lighting: ambient, diffuse, specular, and multiple light sources
- Understand back-face culling and surface shading models

3.1 Color

In a display, a pixel color is specified as a red, green, and blue (RGB) vector. The RGB colors are also called the *primaries*, because our eye sees a different color in a vector of different primary values. The RGB colors are additive primaries — we construct a color on the black background by adding the primaries together. For example, with equal amounts of R, G, and B: G+B ⇒ cyan, R+B ⇒ magenta, R+G ⇒ yellow, and R+G+B ⇒ white. RGB colors are used in the graphics hardware, which we will discuss in more detail.

Cyan, magenta, and yellow (CMY) colors are the complements of RGB colors, respectively. The CMY colors are subtractive primaries — we construct a color on a white background by removing the corresponding RGB primaries. Similarly, with equal amounts of R, G, and B: C = RGB - R, M = RGB - G, and Y = RGB - B.

The CMY colors are used in color printers. Adding certain amounts of CMY inks to a point on a white paper is like removing certain amounts of RGB from the white color at that point. The resulting color at the point on the paper depends on the portions of individual inks. Black ink is used to generate different levels of grays replacing use of equal amounts of CMY inks.
3 Color and Lighting

3.1.1 RGB Mode and Index Mode

If each pixel value in the frame buffer is an RGB vector, the display is in RGB mode. Each pixel value can also be an index into a color look-up table called a colormap, as shown in Fig. 3.1. Then, the display is in index mode. The pixel color is specified in the colormap instead of the frame buffer.

Let’s assume that we have 3 bits per entry in the frame buffer. That is, the frame buffer has 3 bitplanes. In RGB mode, we have access to 8 different colors: black, red, green, blue, cyan, magenta, yellow, and white. In index mode, we still have access to only 8 different colors, but the colors can vary depending on how we load the colormap. If the graphics hardware has a limited number of bitplanes for the frame buffer, index mode allows more color choices, even though the number of colors is the same as that of RGB mode at the same time. For example, in the above example, if we have 12 bitplanes per entry in the colormap, we can choose 8 colors from $2^{12} = 4096$ different colors. The colormap does not take much space in memory, which had been a significant advantage when fast memory chips were very expensive. In GLUT, we use `glutInitDisplayMode(GLUT_INDEX)` to choose the index mode. RGB mode is the default. Index mode can also be useful for doing various animation tricks. However, in general, because memory is no longer a limitation and RGB mode is easier and more flexible, we use it in the examples. Also, in OpenGL programming, each color component (R, G, or B) value is in the range of 0 to 1. The system will scale the value to the corresponding hardware bits during compilation transparent to the users.