Chapter 6

Color and Shading

The perception of color is very important for humans. Color perception depends upon both the physics of the light and complex processing by the eye-brain which integrates properties of the stimulus with experience. Humans use color information to distinguish objects, materials, food, places, and even the time of day. Figure 6.1 shows the same scene coded with different colors: even though all shapes are the same, the right image is quite different from the left and the viewer might interpret it as an indoor scene of a housecat rather than a tiger in grass.

With recent innovation in economical devices, color processing by machine has become commonplace: we have color cameras, color displays and software that processes color images. Color can also be used by machines for the same purposes humans use it. Color is especially convenient because it provides multiple measurements at a single pixel of the image, often enabling classification to be done without complex spatial decision-making.

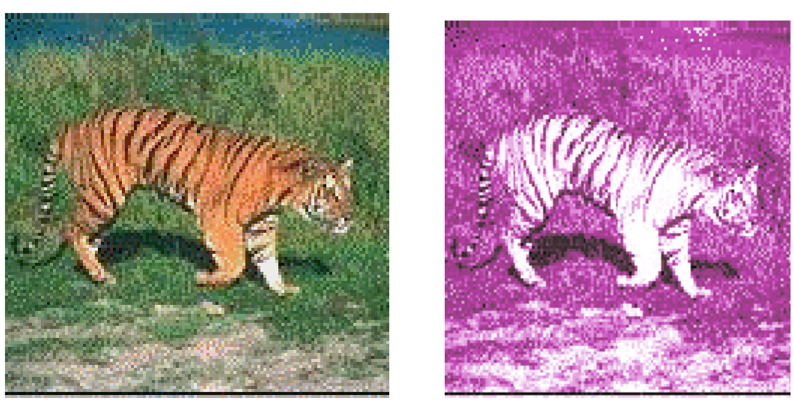

Figure 6.1: (Left) Naturally colored image of tiger in grass; (right) with transformed colors, recognition of a tiger is less secure – perhaps it's a cat on a rug?

Careful study of the physics and perception of color would require many pages: here we provide only a foundation that should be sufficient for beginning programming using color or as a guide to the literature. Some basic principles of the physics of color are given along with practical methods of coding color information in images. Then, we give some examples and methods for using color in recognizing objects and segmenting images.

We also study the *shading* of objects, which depends not just on the color of objects and the light illuminating them, but also on many other factors. These factors include the roughness of surfaces, the angles made between the surface and both the light sources and the viewer, and the distances of the surface from both the light sources and the viewer. Color and shading, important elements of art for centuries, are also important for interpreting a scene by computer vision algorithms.

6.1 Some Physics of Color

Electromagnetic radiation with wavelength λ in the range of between about 400 and 700 nanometers stimulates human neurosensors and produces the sensation of color. (Figure 6.2). A nanometer is 10^{-9} meter: it is also referred to as a millimicron. For blue light, $400 \times 10^{-9} meters\ per\ wave$ means $2.5 \times 10^6 waves\ per\ meter$ or 25000 $waves\ per\ cm$. The speed of light in a vacuum is $3 \times 10^8 m/sec$, which is equivalent to a frequency of 0.75×10^{15} blue light waves per second. This frequency is one one thousandth of that for X-rays and one billion times that of broadcast radio waves.

For the rest of this chapter, we refer to wavelength or frequency only in the context of the qualitative color it produces. Machines can detect radiation well beyond the range of human neurosensors; for example, short ultraviolet waves and extremely short X-rays can be detected by special devices. Also, long infrared waves can be detected by many solid state cameras, and very long radio waves can be detected by a radio receiver. Science and engineering have developed many devices to sense and transduce pixel measurements into the visible spectrum: the X-ray machine and IR satellite weather scanner are two common examples.

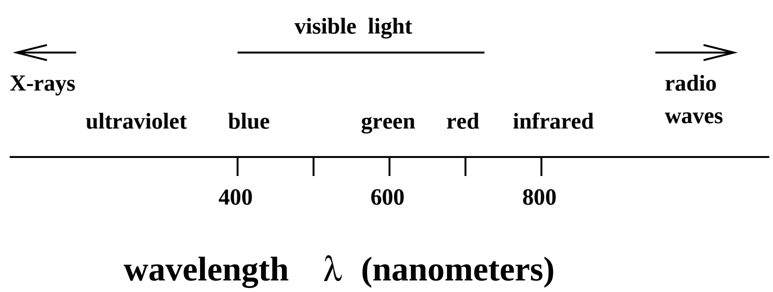

Figure 6.2: Visible part of the electromagnetic spectrum.

Exercise 97

Suppose a piece of paper is 0.004 inches thick. What is its thickness in terms of the equivalent number of waves of blue light?

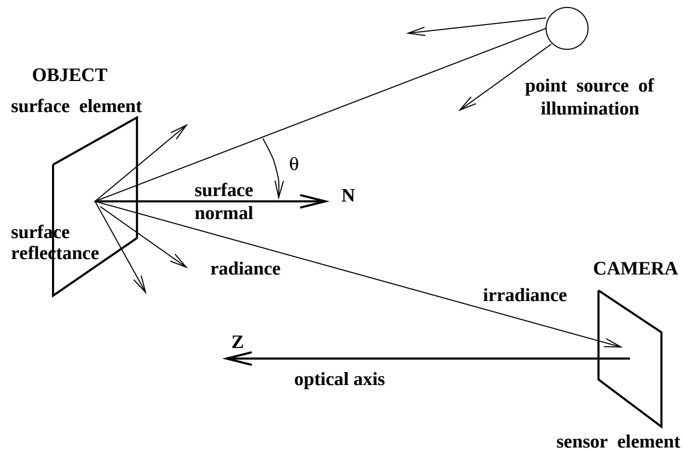

Figure 6.3: Light energy from a source reflects from an object surface and irradiates a sensor element.

6.1.1 Sensing Illuminated Objects

Figure 6.3 shows light from a point source illuminating an object surface. As a result of the illuminating energy interacting with molecules of the object surface, light energy, or radiance, is emitted from the surface, some of which irradiates, or stimulates, a sensor element in a camera or organism's eye. The sensation, or perception, of an object's color depends upon three general factors:

- the spectrum of energy in various wavelengths illuminating the object surface,
- the spectral reflectance of the object surface, which determines how the surface changes the received spectrum into the radiated spectrum,
- the spectral sensitivity of the sensor irradiated by the light energy from the object's surface.

An object that is "blue" has a surface material that appears blue when illuminated with *white light*.

64 DEFINITION **White light** *is composed of approximately equal energy in all wavelengths of the visible spectrum.*

This same object should appear violet if illuminated by only red light. A blue car under intense (white) sunlight will become hot to the touch and radiate energy in the IR range, which cannot be seen by the human eye but can be seen by an IR camera.

6.1.2 Additional Factors

In addition to the three major factors given above, there are several complicating factors in both physics and human perception. Surfaces vary in *specularity*, that is, how much they act like a mirror. *Matte* surfaces reflect energy equally in all directions. The energy or intensity of radiation depends upon distance – surface elements farther from a point source of white light will receive less energy than closer surface elements. The effect is similar between the radiating object and the sensor elements. As a result, image intensities received from the same surface material might be nonuniform due to the nonuniform distances along the imaging rays. The orientation θ of the surface element relative to the source is even more important than distance in determining the energy reflected toward the sensor. These issues are discussed in more detail toward the end of this chapter.

Exercise 98 variation of intensity with distance

Point your computer's camera perpendicularly at a sheet of uniform white paper that is illuminated from an incandescent bulb off to one side. Record the image and study the image intensities. How much variation is there? Is there a systematic decrease of intensity as the distance from some brightest pixel increases?

Exercise 99 variation of intensity with surface normal

Repeat the above experiment using a spherical volleyball rather than a flat sheet of paper. Record the image and study the image intensities. Report on the variations and regularities.

6.1.3 Sensitivity of Receptors

Actual receptors react only to some wavelengths and are more sensitive to certain wavelengths than to others. Figure 6.4 shows sample sensitivity curves. Three of the curves correspond to three different kinds of cones in the human eye containing different chemical pigments sensitive to different wavelengths. The curve marked $human_1$ corresponds to a type of cone that is mildly sensitive to *blue* light between 400 and 500 nm. The curve marked $human_2$ corresponds to cones that are very sensitive to green light and mildly sensitive to shorter wavelengths of blue and longer wavelengths of red. The brain fuses the responses from a local neighborhood of several cones to produce the perception of any visible color. It is somewhat remarkable that only three kinds of receptors are needed to do this, even though there are an infinite number of possible wavelengths of light. Many other seeing animals have only one or two types of receptors and perhaps perceive less rich color as a result. Solid state sensing elements usually have good sensitivity above the range for humans. It's important to remember this, since sometimes as the workday warms up, a machine vision system will see a scene differently from what a human operator sees. This is primarily due to the different sensitivity to IR radiation.

Exercise 100 favorite color

Do you have a *favorite color*? Is so, what is it? Why is it your favorite? Ask 3 other people what their favorite color is. Assuming you have multiple answers, how can you explain it given the known physics of color?

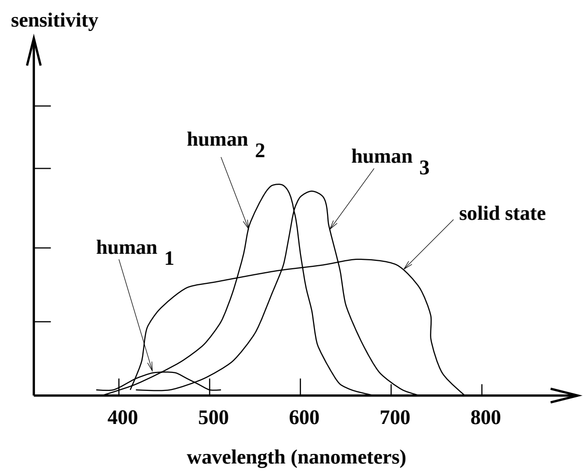

Figure 6.4: Comparison of relative sensitivities of 3 human pigments in cones and solid state sensor element.

6.2 The RGB Basis for Color

Using only three types of receptors, humans can distinguish among thousands of colors; a more exact number is subject to argument. The *trichromatic* RGB (red-green-blue) encoding in graphics systems usually uses three bytes enabling $(2^8)^3$ or roughly 16 million distinct color codes. To be precise, we say 16 million *codes* and not 16 million *colors* because humans cannot actually perceive that many distinct colors. Machines can distinguish between any pair of different bit encodings, but the encodings may or may not represent differences that are significant in the real world. Each 3-byte or 24-bit RGB pixel includes one byte for each of red, green, and blue. The order in which each appears in memory can vary; order is irrelevant to theory but important for programming. Display devices whose color resolution matches the human eye are said to use *true color*. At least 16 bits are needed: a 15-bit encoding might use 5 bits for each of R,B,G, while a 16-bit encoding would better model the relatively larger green sensitivity using 6 bits.

The encoding of an arbitrary color in the visible spectrum can be made by combining the encoding of three *primary colors* (RGB) as shown in Figure 6.5. Red:(255,0,0) and green:(0,255,0) combined in equal amounts create yellow:(255,255,0). The amount of each primary color gives its intensity. If all components are of highest intensity, then the color white results. Equal proportions of less intensity create shades of grey:(c,c,c) for any constant $0 < c < 255$ down to black:(0,0,0). It is often more convenient to scale values in the range 0 to 1 rather than 0 to 255 when making decisions about color in our algorithms: use of such a range is *device-independent*.

The RGB system is an *additive color system* because colors are created by adding components to black:(0,0,0). This corresponds well to RGB displays (monitors) which have three types of phosphors to emit light. Three neighboring elements of phosphor corresponding to a pixel are struck by three electron beams of intensity c_1, c_2 and c_3 respectively: the human

	RGB	CMY	HSI
RED	(255, 0, 0)	(0,255,255)	(0.0 , 1.0, 255)
YELLOW	(255,255, 0)	(0, 0,255)	(1.05, 1.0, 255)
	(100,100, 50)	(155,155,205)	(1.05, 0.5, 100)
GREEN	(0,255, 0)	(255, 0,255)	(2.09, 1.0, 255)
BLUE	(0, 0,255)	(255,255, 0)	(4.19, 1.0, 255)
WHITE	(255,255,255)	(0, 0, 0)	(-1.0, 0.0, 255)
GREY	(192,192,192)	(63, 63, 63)	(-1.0, 0.0, 192)
	(127,127,127)	(128,128,128)	(-1.0, 0.0, 127)
	(63, 63, 63)	(192,192,192)	(-1.0, 0.0, 63)
	...		
BLACK	(0, 0, 0)	(255,255,255)	(-1.0, 0.0, 0)

Figure 6.5: Different digital trichromatic color encoding systems. It is often more convenient to scale values in the range 0 to 1 when making decisions in algorithms. HSI values are computed from RGB values using Algorithm 15: $H \in [0.0, 2\pi)$, $S \in [0.0, 1.0]$ and $I \in [0, 255]$. Byte codings exist for H and S.

eye integrates their luminance to perceive "color":(c_1, c_2, c_3). The light of 3 wavelengths from a small region of the CRT screen is thus physically added or mixed together.

Suppose that a color sensor encodes a pixel of a digital image as (R, G, B), where each coordinate is in the range $[0, 255]$, for example. The computations shown in Equation 6.1 are one way to normalize image data for interpretation by both computer programs and people and for transformation to other color systems as discussed below. Imagine a color camera staring at a scene with variations in illumination; for example, object surface points are at varying distances from illumination sources and may even be in shadow relative to some of the light sources. An algorithm to aggregate green pixels corresponding to the image of a car would perform poorly unless the normalization for intensity were done first.

$$\begin{aligned} intensity\ I &= (R + G + B)/3 \\ normalized\ red\ r &= R/(R + G + B) \\ normalized\ green\ g &= G/(R + G + B) \\ normalized\ blue\ b &= B/(R + G + B) \end{aligned} \tag{6.1}$$

Using the normalization of Equation 6.1, the normalized values will always sum to 1. There are alternative normalizations; for instance, we could use $max(R, G, B)$ as the divisor rather than the average RBG value. By using $r + g + b = 1$, the relationship of coordinate values to colors can be conveniently plotted via a 2D graph as in Figure 6.6. Pure colors

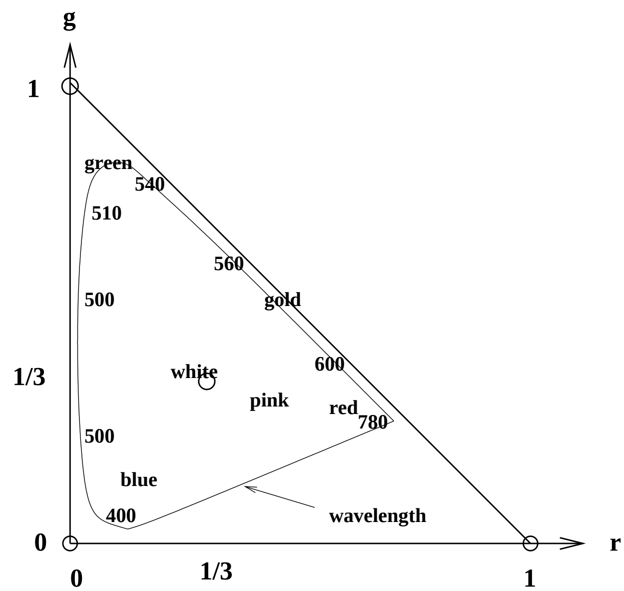

Figure 6.6: Color triangle for normalized RGB coordinates. The blue ('b') axis is out of the page perpendicular to the 'r' and 'g' axes. Thus, the triangle is actually a slice through the points [1,0,0], [0,1,0] and [0,0,1] in 3D. The value for blue can be computed as $b = 1 - r - g$ for any pair of r-g values shown in the triangle.

are represented by points near the corners of the triangle. For example, a "fire-engine-red" will be near the lower right corner with coordinates (1,0) and a "grass-green" will be at the top with coordinates (0,1) while "white" will be at the centroid (1/3,1/3). In Figure 6.6, the blue ('b') axis is out of the page perpendicular to the 'r' and 'g' axes, and thus the triangle is actually a slice through the points [1,0,0], [0,1,0] and [0,0,1] in 3D. The value for blue can be computed as $b = 1 - r - g$ for any pair of r-g values shown inside the triangle.

Exercise 101 experimenting with color codes

Aquire an RGB color image and view it with some image tool. Exchange the green and blue bytes and report on the results. Double all and only the low blue values and report on the results.

6.3 Other Color Bases

Several other color bases exist which have special advantages relative to devices that produce color or relative to human perception. Some bases are merely linear transformations of others and some are not.

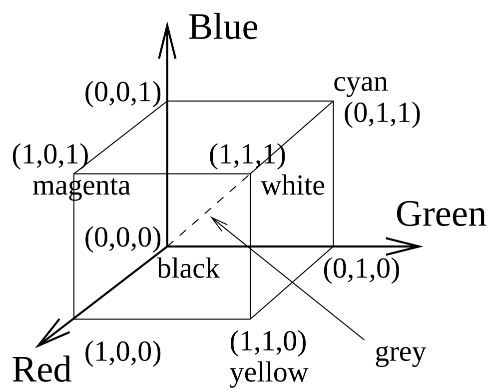

Figure 6.7: Color cube for normalized RGB coordinates: the triangle in Figure 6.6 is a projection of the plane through points [1, 0, 0], [0, 1, 0], and [0, 0, 1].

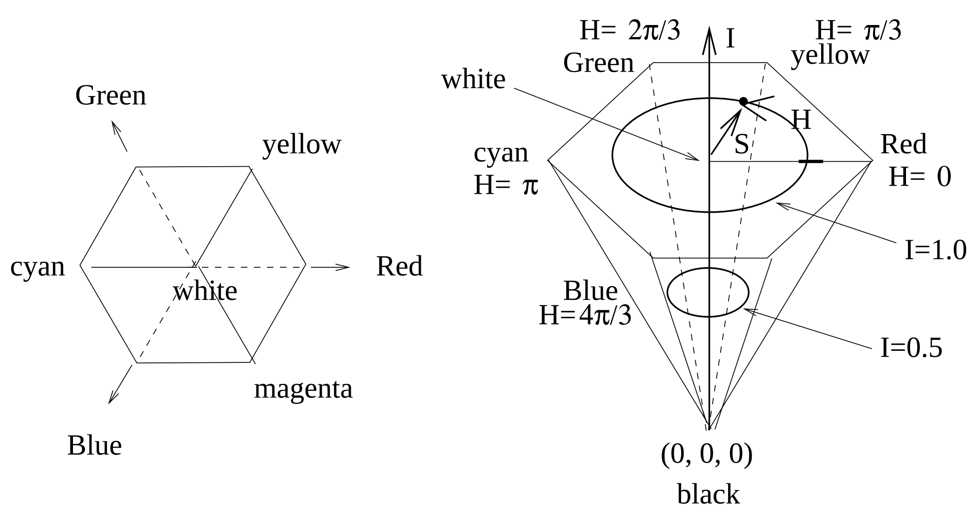

Figure 6.8: Color hexacone for HSI representation. At the left is a projection of the RGB cube perpendicular to the diagonal from (0, 0, 0) to (1, 1, 1): color names now appear at the vertices of a hexagon. At the right is a hexacone representing colors in HSI coordinates: intensity (I) is the vertical axis; hue (H) is an angle from 0 to 2π with *RED* at 0.0; saturation (S) ranges from 0 to 1 according to how pure, or unlike white, the color is with S=0.0 corresponding to the **I**-axis.

6.3.1 The CMY Subtractive Color System

The CMY color system models printing on white paper and *subtracts from white* rather than adds to black as the RGB system does. CMY coding is shown next to RGB in Figure 6.5. CMY is an abbreviation of *Cyan-Magenta-Yellow*, which are its three primary colors corresponding to three inks. Cyan absorbs red illumination, magenta absorbs green and yellow absorbs blue, thus creating appropriate reflections when the printed image is illuminated with white light. The system is termed *subtractive* because of the encoding for absorption. Some trichromatic encodings are as follows; white:(0,0,0) because no white illumination should be absorbed, black:(255,255,255) because all components of white light should be absorbed and yellow:(0,0,255) because the blue component of incident white light should be absorbed by the inks, leaving the red and green components to create the perception of yellow.

6.3.2 HSI: Hue-Saturation-Intensity

The HSI system encodes color information by separating out an overall intensity value I from two values encoding *chromaticity* — hue H and saturation S. The color cube in Figure 6.7 is related to the RGB triangle shown in Figure 6.6. In the cube representation, each r, g, b value can range independently in $[0.0, 1.0]$. If we project the color cube along its major diagonal, we arrive at the hexagon at the left of Figure 6.8. In this representation, shades of grey that were formerly along the color cube diagonal now are all projected to the center "white" point while the "red" point [1,0,0] is now at the right corner and the green point [0,1,0] is at the top left corner of the hexagon. A related 3D representation, called a "hexacone", is shown at the right in Figure 6.8: the 3D representation allows us to visualize the former cube diagonal as a vertical intensity axis **I**. Hue **H** is defined by an *angle* between 0 and 2π relative to the "red"-axis, with pure "red" at an angle of 0, pure "green" at $2\pi/3$ and pure "blue" at $4\pi/3$. Saturation **S** is the 3rd coordinate value needed in order to completely specify a point in this color space. Saturation models the purity of the color or hue, with 1 modeling a completely pure or saturated color and 0 modeling a completely unsaturated hue, i.e. some shade of "grey".

The HSI system is sometimes referred to as the "HSV" system using the term "value" instead of "intensity". HSI is more convenient to some graphics designers because it provides direct control of brightness and hue. Pastels are centered near the **I** axis, while deep or rich colors are out at the periphery of the hexacone. HSI might also provide better support for computer vision algorithms because it can normalize for lighting and focus on the two chromaticity parameters that are more associated with the intrinsic character of a surface rather than the source that is lighting it.

Derivation of HSI coordinates from RGB coordinates is given in Algorithm 15. The algorithm can convert input values (r, g, b) from the 3D color cube, or those normalized by Equation 6.1, or even byte-coded RGB values as in the left column of Figure 6.5. Intensity I is returned in the same range as the input values. Saturation S is not defined when intensity $I = 0$ and hue H is not defined when $S = 0$. H is in the range $[0,\ 2\pi)$. Whereas one might use a square root and inverse cosine to define mathematical conversion formulas, Algorithm 15 uses very simple computational operations so that it will run fast when converting an entire image of pixels from one encoding to another. Samples of the output of Algorithm 15 are given at the right in Figure 6.5.

Conversion of RGB encoding to HSI encoding.

R,G,B : input values of RGB all in range [0,1] or [0,255];
I : output value of intensity in same range as input;
S : output value of saturation in range [0,1];
H : output value of hue in range $[0,2\pi)$, -1 if S is 0;
R,G,B,H,S,I are all floating point numbers;

```
procedure RGB_to_HSI( in R,G,B; out H,S,I)
{
I := max ( R, G, B );
min := min ( R, G, B );
if (I ≥ 0.0) then S := (I - min )/I else S := 0.0;
if (S ≤ 0.0) then { H := -1.0; return; }
   "compute the hue based on the relative sizes of the RGB components"
diff := I - min;
"is the point within +/- 60 degrees of the red axis?"
if (r = I) then H := (π/3)*(g - b)/diff;
"is the point within +/- 60 degrees of the green axis?"
else if (g = I) then H := (2 * π/3) + π/3 *(b - r)/diff;
"is the point within +/- 60 degrees of the blue axis?"
else if (b = I) then H := (4 * π/3) + π/3 *(r - g)/diff;
if (H ≤ 0.0) H := H + 2π;
}
```

Algorithm 15: Conversion of RGB to HSI.

Exercise 102

Using Algorithm 15, (a) convert the RGB code (100,150,200) into an HSI code and (b) convert the rgb code (0.0, 1.0, 0.0) to HSI.

Returning to Figure 6.6, we see how HSI values relate to the color triangle. Hue is related to the dominant wavelength of the light and corresponds approximately to a point on the sides of the triangle in Figure 6.6 with the lower values of λ near 400 nm starting at the origin and increasing along the $g - axis$ to about 520 nm and further increasing toward 800 nm down along the hypotenuse. Hue corresponds to the angle from the centroid corresponding to "white" toward some point (r, g) on a side of the triangle. The H and S values for 50% saturated gold is midway between the points marked "white" and "gold" in Figure 6.6. Figure 6.6 is an approximation to the *painters color palette.*

Figure 6.9 shows the transformation of an image by changing its saturation. The original input image is at the left. The center image is the result of decreasing the saturation S of all individual pixels by 20% and the right image is the result of a 20% increase in S. Relative