

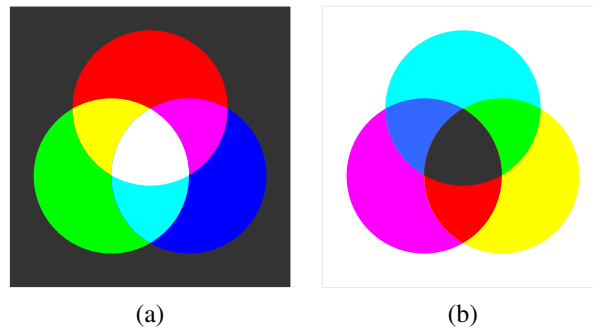
**Figure 2.26** Sample point spread functions (PSF): The diameter of the blur disc (blue) in (a) is equal to half the pixel spacing, while the diameter in (c) is twice the pixel spacing. The horizontal fill factor of the sensing chip is 80% and is shown in brown. The convolution of these two kernels gives the point spread function, shown in green. The Fourier response of the PSF (the MTF) is plotted in (b) and (d). The area above the Nyquist frequency where aliasing occurs is shown in red.

ious image processing operations, such as resampling, upsampling, and downsampling. Sections 3.4 and 3.5.2 discuss these issues and show how careful selection of filters can reduce the amount of aliasing that operations inject.

### 2.3.2 Color

In Section 2.2, we saw how lighting and surface reflections are functions of wavelength. When the incoming light hits the imaging sensor, light from different parts of the spectrum is somehow integrated into the discrete red, green, and blue (RGB) color values that we see in a digital image. How does this process work and how can we analyze and manipulate color values?

You probably recall from your childhood days the magical process of mixing paint colors to obtain new ones. You may recall that blue+yellow makes green, red+blue makes purple, and red+green makes brown. If you revisited this topic at a later age, you may have learned that the proper *subtractive* primaries are actually cyan (a light blue-green), magenta (pink), and yellow (Figure 2.27b), although black is also often used in four-color printing (CMYK). (If you ever subsequently took any painting classes, you learned that colors can have even more fanciful names, such as alizarin crimson, cerulean blue, and chartreuse.) The subtractive colors are called subtractive because pigments in the paint absorb certain wavelengths in the color spectrum.



**Figure 2.27** Primary and secondary colors: (a) additive colors red, green, and blue can be mixed to produce cyan, magenta, yellow, and white; (b) subtractive colors cyan, magenta, and yellow can be mixed to produce red, green, blue, and black.

Later on, you may have learned about the *additive* primary colors (red, green, and blue) and how they can be added (with a slide projector or on a computer monitor) to produce cyan, magenta, yellow, white, and all the other colors we typically see on our TV sets and monitors (Figure 2.27a).

Through what process is it possible for two different colors, such as red and green, to interact to produce a third color like yellow? Are the wavelengths somehow mixed up to produce a new wavelength?

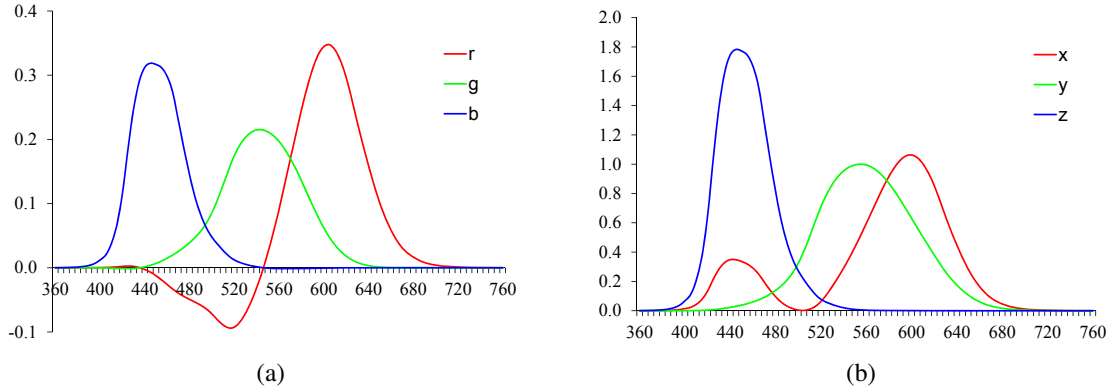
You probably know that the correct answer has nothing to do with physically mixing wavelengths. Instead, the existence of three primaries is a result of the *tri-stimulus* (or *tri-chromatic*) nature of the human visual system, since we have three different kinds of cone, each of which responds selectively to a different portion of the color spectrum (Glassner 1995; Wyszecki and Stiles 2000; Fairchild 2005; Reinhard, Ward, Pattanaik *et al.* 2005; Livingstone 2008).<sup>18</sup> Note that for machine vision applications, such as remote sensing and terrain classification, it is preferable to use many more wavelengths. Similarly, surveillance applications can often benefit from sensing in the near-infrared (NIR) range.

### CIE RGB and XYZ

To test and quantify the tri-chromatic theory of perception, we can attempt to reproduce all *monochromatic* (single wavelength) colors as a mixture of three suitably chosen primaries. (Pure wavelength light can be obtained using either a prism or specially manufactured color filters.) In the 1930s, the Commission Internationale d’Eclairage (CIE) standardized the RGB representation by performing such *color matching* experiments using the primary colors of red (700.0nm wavelength), green (546.1nm), and blue (435.8nm).

Figure 2.28 shows the results of performing these experiments with a *standard observer*, i.e., averaging perceptual results over a large number of subjects. You will notice that for certain pure spectra in the blue–green range, a *negative* amount of red light has to be added, i.e., a certain amount of red has to be added to the color being matched in order to get a color match. These results also provided a simple explanation for the existence of *metamers*, which are colors with different spectra that are perceptually indistinguishable. Note that two fabrics

<sup>18</sup> See also Mark Fairchild’s Web page, [http://www.cis.rit.edu/fairchild/WhyIsColor/books\\_links.html](http://www.cis.rit.edu/fairchild/WhyIsColor/books_links.html).



**Figure 2.28** Standard CIE color matching functions: (a)  $\bar{r}(\lambda)$ ,  $\bar{g}(\lambda)$ ,  $\bar{b}(\lambda)$  color spectra obtained from matching pure colors to the R=700.0nm, G=546.1nm, and B=435.8nm primaries; (b)  $\bar{x}(\lambda)$ ,  $\bar{y}(\lambda)$ ,  $\bar{z}(\lambda)$  color matching functions, which are linear combinations of the  $(\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda))$  spectra.

or paint colors that are metamers under one light may no longer be so under different lighting.

Because of the problem associated with mixing negative light, the CIE also developed a new color space called XYZ, which contains all of the pure spectral colors within its positive octant. (It also maps the Y axis to the *luminance*, i.e., perceived relative brightness, and maps pure white to a diagonal (equal-valued) vector.) The transformation from RGB to XYZ is given by

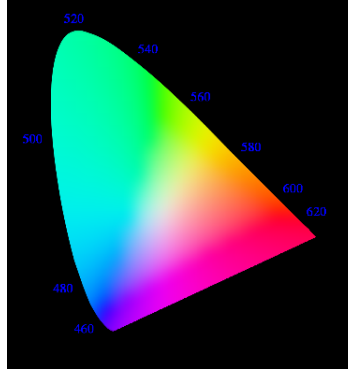
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0.17697} \begin{bmatrix} 0.49 & 0.31 & 0.20 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00 & 0.01 & 0.99 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (2.103)$$

While the official definition of the CIE XYZ standard has the matrix normalized so that the Y value corresponding to pure red is 1, a more commonly used form is to omit the leading fraction, so that the second row adds up to one, i.e., the RGB triplet (1, 1, 1) maps to a Y value of 1. Linearly blending the  $(\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda))$  curves in Figure 2.28a according to (2.103), we obtain the resulting  $(\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda))$  curves shown in Figure 2.28b. Notice how all three spectra (color matching functions) now have only positive values and how the  $\bar{y}(\lambda)$  curve matches that of the luminance perceived by humans.

If we divide the XYZ values by the sum of X+Y+Z, we obtain the *chromaticity coordinates*

$$x = \frac{X}{X + Y + Z}, \quad y = \frac{Y}{X + Y + Z}, \quad z = \frac{Z}{X + Y + Z}, \quad (2.104)$$

which sum up to 1. The chromaticity coordinates discard the absolute intensity of a given color sample and just represent its pure color. If we sweep the monochromatic color  $\lambda$  parameter in Figure 2.28b from  $\lambda = 380\text{nm}$  to  $\lambda = 800\text{nm}$ , we obtain the familiar *chromaticity diagram* shown in Figure 2.29. This figure shows the  $(x, y)$  value for every color value perceivable by most humans. (Of course, the CMYK reproduction process in this book does not actually span the whole gamut of perceivable colors.) The outer curved rim represents where



**Figure 2.29** CIE chromaticity diagram, showing colors and their corresponding  $(x, y)$  values. Pure spectral colors are arranged around the outside of the curve.

all of the pure monochromatic color values map in  $(x, y)$  space, while the lower straight line, which connects the two endpoints, is known as the *purple line*.

A convenient representation for color values, when we want to tease apart luminance and chromaticity, is therefore Yxy (luminance plus the two most distinctive chrominance components).

### L\*a\*b\* color space

While the XYZ color space has many convenient properties, including the ability to separate luminance from chrominance, it does not actually predict how well humans perceive *differences* in color or luminance.

Because the response of the human visual system is roughly logarithmic (we can perceive *relative* luminance differences of about 1%), the CIE defined a non-linear re-mapping of the XYZ space called L\*a\*b\* (also sometimes called CIELAB), where differences in luminance or chrominance are more perceptually uniform.<sup>19</sup>

The L\* component of *lightness* is defined as

$$L^* = 116f\left(\frac{Y}{Y_n}\right), \quad (2.105)$$

where  $Y_n$  is the luminance value for nominal white (Fairchild 2005) and

$$f(t) = \begin{cases} t^{1/3} & t > \delta^3 \\ t/(3\delta^2) + 2\delta/3 & \text{else,} \end{cases} \quad (2.106)$$

is a finite-slope approximation to the cube root with  $\delta = 6/29$ . The resulting 0 . . . 100 scale roughly measures equal amounts of lightness perceptibility.

In a similar fashion, the a\* and b\* components are defined as

$$a^* = 500 \left[ f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right] \quad \text{and} \quad b^* = 200 \left[ f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right], \quad (2.107)$$

<sup>19</sup> Another perceptually motivated color space called L\*u\*v\* was developed and standardized simultaneously (Fairchild 2005).

where again,  $(X_n, Y_n, Z_n)$  is the measured white point. Figure 2.32i–k show the  $L^*a^*b^*$  representation for a sample color image.

### Color cameras

While the preceding discussion tells us how we can uniquely describe the perceived tri-stimulus description of any color (spectral distribution), it does not tell us how RGB still and video cameras actually work. Do they just measure the amount of light at the nominal wavelengths of red (700.0nm), green (546.1nm), and blue (435.8nm)? Do color monitors just emit exactly these wavelengths and, if so, how can they emit negative red light to reproduce colors in the cyan range?

In fact, the design of RGB video cameras has historically been based around the availability of colored phosphors that go into television sets. When standard-definition color television was invented (NTSC), a mapping was defined between the RGB values that would drive the three color guns in the cathode ray tube (CRT) and the XYZ values that unambiguously define perceived color (this standard was called ITU-R BT.601). With the advent of HDTV and newer monitors, a new standard called ITU-R BT.709 was created, which specifies the XYZ values of each of the color primaries,

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R_{709} \\ G_{709} \\ B_{709} \end{bmatrix}. \quad (2.108)$$

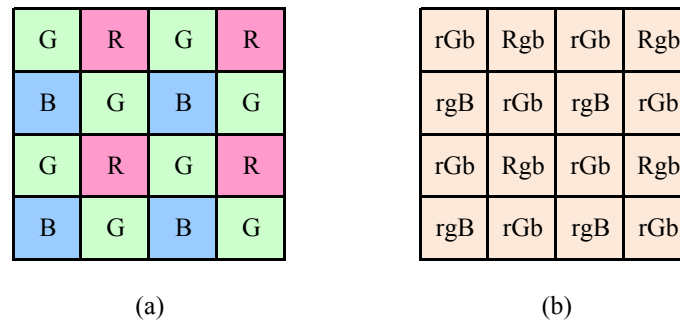
In practice, each color camera integrates light according to the *spectral response function* of its red, green, and blue sensors,

$$\begin{aligned} R &= \int L(\lambda) S_R(\lambda) d\lambda, \\ G &= \int L(\lambda) S_G(\lambda) d\lambda, \\ B &= \int L(\lambda) S_B(\lambda) d\lambda, \end{aligned} \quad (2.109)$$

where  $L(\lambda)$  is the incoming spectrum of light at a given pixel and  $\{S_R(\lambda), S_G(\lambda), S_B(\lambda)\}$  are the red, green, and blue *spectral sensitivities* of the corresponding sensors.

Can we tell what spectral sensitivities the cameras actually have? Unless the camera manufacturer provides us with this data or we observe the response of the camera to a whole spectrum of monochromatic lights, these sensitivities are *not* specified by a standard such as BT.709. Instead, all that matters is that the tri-stimulus values for a given color produce the specified RGB values. The manufacturer is free to use sensors with sensitivities that do not match the standard XYZ definitions, so long as they can later be converted (through a linear transform) to the standard colors.

Similarly, while TV and computer monitors are supposed to produce RGB values as specified by Equation (2.108), there is no reason that they cannot use digital logic to transform the incoming RGB values into different signals to drive each of the color channels. Properly calibrated monitors make this information available to software applications that perform *color management*, so that colors in real life, on the screen, and on the printer all match as closely as possible.



**Figure 2.30** Bayer RGB pattern: (a) color filter array layout; (b) interpolated pixel values, with unknown (guessed) values shown as lower case.

### Color filter arrays

While early color TV cameras used three *vidicons* (tubes) to perform their sensing and later cameras used three separate RGB sensing chips, most of today's digital still and video cameras use a *color filter array* (CFA), where alternating sensors are covered by different colored filters.<sup>20</sup>

The most commonly used pattern in color cameras today is the *Bayer pattern* (Bayer 1976), which places green filters over half of the sensors (in a checkerboard pattern), and red and blue filters over the remaining ones (Figure 2.30). The reason that there are twice as many green filters as red and blue is because the luminance signal is mostly determined by green values and the visual system is much more sensitive to high frequency detail in luminance than in chrominance (a fact that is exploited in color image compression—see Section 2.3.3). The process of *interpolating* the missing color values so that we have valid RGB values for all the pixels is known as *demosaicing* and is covered in detail in Section 10.3.1.

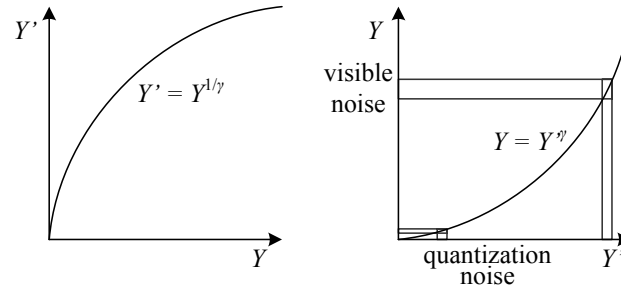
Similarly, color LCD monitors typically use alternating stripes of red, green, and blue filters placed in front of each liquid crystal active area to simulate the experience of a full color display. As before, because the visual system has higher resolution (acuity) in luminance than chrominance, it is possible to digitally pre-filter RGB (and monochrome) images to enhance the perception of crispness (Betrissey, Blinn, Dresevic *et al.* 2000; Platt 2000).

### Color balance

Before encoding the sensed RGB values, most cameras perform some kind of *color balancing* operation in an attempt to move the white point of a given image closer to pure white (equal RGB values). If the color system and the illumination are the same (the BT.709 system uses the daylight illuminant  $D_{65}$  as its reference white), the change may be minimal. However, if the illuminant is strongly colored, such as incandescent indoor lighting (which generally results in a yellow or orange hue), the compensation can be quite significant.

A simple way to perform color correction is to multiply each of the RGB values by a different factor (i.e., to apply a diagonal matrix transform to the RGB color space). More

<sup>20</sup> A newer chip design by Foveon (<http://www.foveon.com>) stacks the red, green, and blue sensors beneath each other, but it has not yet gained widespread adoption.



**Figure 2.31** Gamma compression: (a) The relationship between the input signal luminance  $Y$  and the transmitted signal  $Y'$  is given by  $Y' = Y^{1/\gamma}$ . (b) At the receiver, the signal  $Y'$  is exponentiated by the factor  $\gamma$ ,  $\hat{Y} = Y'^{\gamma}$ . Noise introduced during transmission is squashed in the dark regions, which corresponds to the more noise-sensitive region of the visual system.

complicated transforms, which are sometimes the result of mapping to XYZ space and back, actually perform a *color twist*, i.e., they use a general  $3 \times 3$  color transform matrix.<sup>21</sup> Exercise 2.9 has you explore some of these issues.

### Gamma

In the early days of black and white television, the phosphors in the CRT used to display the TV signal responded non-linearly to their input voltage. The relationship between the voltage and the resulting brightness was characterized by a number called *gamma* ( $\gamma$ ), since the formula was roughly

$$B = V^{\gamma}, \quad (2.110)$$

with a  $\gamma$  of about 2.2. To compensate for this effect, the electronics in the TV camera would pre-map the sensed luminance  $Y$  through an inverse gamma,

$$Y' = Y^{\frac{1}{\gamma}}, \quad (2.111)$$

with a typical value of  $\frac{1}{\gamma} = 0.45$ .

The mapping of the signal through this non-linearity before transmission had a beneficial side effect: noise added during transmission (remember, these were analog days!) would be reduced (after applying the gamma at the receiver) in the darker regions of the signal where it was more visible (Figure 2.31).<sup>22</sup> (Remember that our visual system is roughly sensitive to relative differences in luminance.)

When color television was invented, it was decided to separately pass the red, green, and blue signals through the same gamma non-linearity before combining them for encoding. Today, even though we no longer have analog noise in our transmission systems, signals are still quantized during compression (see Section 2.3.3), so applying inverse gamma to sensed values is still useful.

<sup>21</sup> Those of you old enough to remember the early days of color television will naturally think of the *hue* adjustment knob on the television set, which could produce truly bizarre results.

<sup>22</sup> A related technique called *companding* was the basis of the Dolby noise reduction systems used with audio tapes.

Unfortunately, for both computer vision and computer graphics, the presence of gamma in images is often problematic. For example, the proper simulation of radiometric phenomena such as shading (see Section 2.2 and Equation (2.87)) occurs in a linear radiance space. Once all of the computations have been performed, the appropriate gamma should be applied before display. Unfortunately, many computer graphics systems (such as shading models) operate directly on RGB values and display these values directly. (Fortunately, newer color imaging standards such as the 16-bit sRGB use a linear space, which makes this less of a problem (Glassner 1995).)

In computer vision, the situation can be even more daunting. The accurate determination of surface normals, using a technique such as photometric stereo (Section 12.1.1) or even a simpler operation such as accurate image deblurring, require that the measurements be in a linear space of intensities. Therefore, it is imperative when performing detailed quantitative computations such as these to first undo the gamma and the per-image color re-balancing in the sensed color values. Chakrabarti, Scharstein, and Zickler (2009) develop a sophisticated 24-parameter model that is a good match to the processing performed by today's digital cameras; they also provide a database of color images you can use for your own testing.<sup>23</sup>

For other vision applications, however, such as feature detection or the matching of signals in stereo and motion estimation, this linearization step is often not necessary. In fact, determining whether it is necessary to undo gamma can take some careful thinking, e.g., in the case of compensating for exposure variations in image stitching (see Exercise 2.7).

If all of these processing steps sound confusing to model, they are. Exercise 2.10 has you try to tease apart some of these phenomena using empirical investigation, i.e., taking pictures of color charts and comparing the RAW and JPEG compressed color values.

### Other color spaces

While RGB and XYZ are the primary color spaces used to describe the spectral content (and hence tri-stimulus response) of color signals, a variety of other representations have been developed both in video and still image coding and in computer graphics.

The earliest color representation developed for video transmission was the YIQ standard developed for NTSC video in North America and the closely related YUV standard developed for PAL in Europe. In both of these cases, it was desired to have a *luma* channel Y (so called since it only roughly mimics true luminance) that would be comparable to the regular black-and-white TV signal, along with two lower frequency *chroma* channels.

In both systems, the Y signal (or more appropriately, the Y' luma signal since it is gamma compressed) is obtained from

$$Y'_{601} = 0.299R' + 0.587G' + 0.114B', \quad (2.112)$$

where R'G'B' is the triplet of gamma-compressed color components. When using the newer color definitions for HDTV in BT.709, the formula is

$$Y'_{709} = 0.2125R' + 0.7154G' + 0.0721B'. \quad (2.113)$$

The UV components are derived from scaled versions of  $(B' - Y')$  and  $(R' - Y')$ , namely,

$$U = 0.492111(B' - Y') \text{ and } V = 0.877283(R' - Y'), \quad (2.114)$$

<sup>23</sup> <http://vision.middlebury.edu/color/>.



whereas the IQ components are the UV components rotated through an angle of  $33^\circ$ . In composite (NTSC and PAL) video, the chroma signals were then low-pass filtered horizontally before being modulated and superimposed on top of the Y' luma signal. Backward compatibility was achieved by having older black-and-white TV sets effectively ignore the high-frequency chroma signal (because of slow electronics) or, at worst, superimposing it as a high-frequency pattern on top of the main signal.

While these conversions were important in the early days of computer vision, when frame grabbers would directly digitize the composite TV signal, today all digital video and still image compression standards are based on the newer YCbCr conversion. YCbCr is closely related to YUV (the  $C_b$  and  $C_r$  signals carry the blue and red color difference signals and have more useful mnemonics than UV) but uses different scale factors to fit within the eight-bit range available with digital signals.

For video, the Y' signal is re-scaled to fit within the  $[16 \dots 235]$  range of values, while the Cb and Cr signals are scaled to fit within  $[16 \dots 240]$  (Gomes and Velho 1997; Fairchild 2005). For still images, the JPEG standard uses the full eight-bit range with no reserved values,

$$\begin{bmatrix} Y' \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}, \quad (2.115)$$

where the R'G'B' values are the eight-bit gamma-compressed color components (i.e., the actual RGB values we obtain when we open up or display a JPEG image). For most applications, this formula is not that important, since your image reading software will directly provide you with the eight-bit gamma-compressed R'G'B' values. However, if you are trying to do careful image deblocking (Exercise 3.30), this information may be useful.

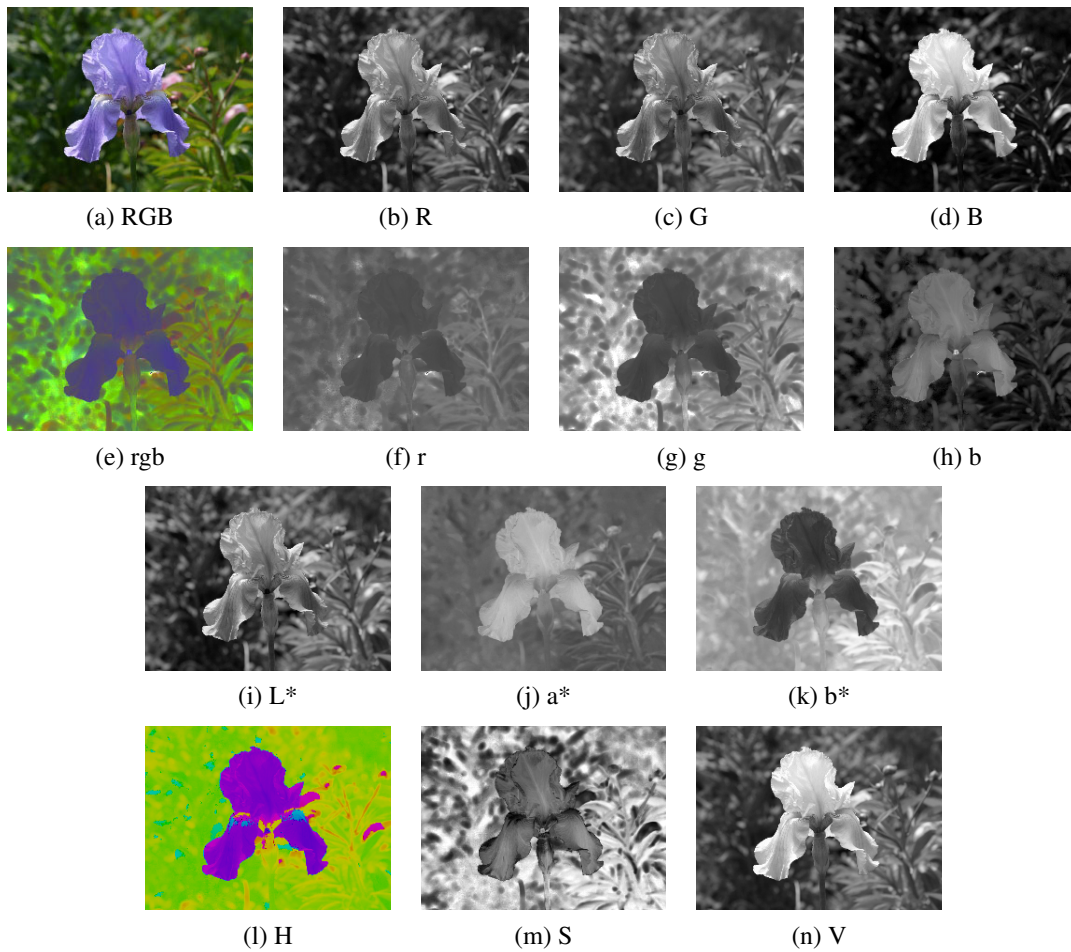
Another color space you may come across is *hue, saturation, value* (HSV), which is a projection of the RGB color cube onto a non-linear chroma angle, a radial saturation percentage, and a luminance-inspired value. In more detail, value is defined as either the mean or maximum color value, saturation is defined as scaled distance from the diagonal, and hue is defined as the direction around a color wheel (the exact formulas are described by Hall (1989); Foley, van Dam, Feiner *et al.* (1995)). Such a decomposition is quite natural in graphics applications such as color picking (it approximates the Munsell chart for color description). Figure 2.32l–n shows an HSV representation of a sample color image, where saturation is encoded using a gray scale (saturated = darker) and hue is depicted as a color.

If you want your computer vision algorithm to only affect the value (luminance) of an image and not its saturation or hue, a simpler solution is to use either the  $Yxy$  (luminance + chromaticity) coordinates defined in (2.104) or the even simpler *color ratios*,

$$r = \frac{R}{R + G + B}, \quad g = \frac{G}{R + G + B}, \quad b = \frac{B}{R + G + B} \quad (2.116)$$

(Figure 2.32e–h). After manipulating the luma (2.112), e.g., through the process of histogram equalization (Section 3.1.4), you can multiply each color ratio by the ratio of the new to old luma to obtain an adjusted RGB triplet.

While all of these color systems may sound confusing, in the end, it often may not matter that much which one you use. Poynton, in his *Color FAQ*, <http://www.poynton.com/ColorFAQ.html>, notes that the perceptually motivated L\*a\*b\* system is qualitatively similar



**Figure 2.32** Color space transformations: (a–d) RGB; (e–h) rgb. (i–k)  $L^*a^*b^*$ ; (l–n) HSV. Note that the rgb,  $L^*a^*b^*$ , and HSV values are all re-scaled to fit the dynamic range of the printed page.

to the gamma-compressed  $R'G'B'$  system we mostly deal with, since both have a fractional power scaling (which approximates a logarithmic response) between the actual intensity values and the numbers being manipulated. As in all cases, think carefully about what you are trying to accomplish before deciding on a technique to use.<sup>24</sup>

### 2.3.3 Compression

The last stage in a camera's processing pipeline is usually some form of image compression (unless you are using a lossless compression scheme such as camera RAW or PNG).

All color video and image compression algorithms start by converting the signal into YCbCr (or some closely related variant), so that they can compress the luminance signal with

<sup>24</sup> If you are at a loss for questions at a conference, you can always ask why the speaker did not use a perceptual color space, such as  $L^*a^*b^*$ . Conversely, if they did use  $L^*a^*b^*$ , you can ask if they have any concrete evidence that this works better than regular colors.