The current standard, since 2004, for specifying the color properties of a device is the **ICC profile** established by the **International Color Consortium**. Three different types of devices can have ICC profiles: input devices (e.g., cameras or scanners), display devices (e.g., computer monitors), and output devices (e.g., printers). Manufacturers carefully calibrate their devices so that they are shipped with ICC profiles; in this way, various input and output devices can be connected without knowing about each other, thus facilitating accurate color reproduction. An ICC profile specifies the transformation from the values in the color space of the device to one of two standard color spaces, either CIE 1931 *XYZ* or CIELAB (defined later), where the transformation takes the form of either a 3×3 matrix or a lookup table (LUT) with values for different wavelengths. The profile corresponds to a reference viewing condition and the standard illuminant D_{50}. Each profile also includes one of four rendering intents: absolute colorimetric, relative colorimetric, perceptual, or saturation. The first two preserve, respectively, either the absolute colorimetry of in-gamut colors, or the colorimetry of in-gamut colors relative to the white point. The latter two compromise the colorimetry of in-gamut colors in order to better handle out-of-gamut colors, with perceptual intent preserving the full range of colors, and saturation intent preserving saturated colors for displaying computer graphics.

9.4.5 CAMs and CATs

The conclusion that color perception can be adequately described using three dimensions is supported by color matching experiments conducted in a carefully controlled viewing environment with a limited field of view, no context, and a particular state of adaptation. In real-world viewing, with different illumination levels and wider fields of view, however, more complex models are necessary. Such models are known as **color appearance models (CAMs)**, with the most recent standard being **CIECAM02**. In a CAM, the visual perception of color appearance is defined by 6 dimensions, including 3 that describe the object properties (lightness, chroma, and hue) and 3 that describe the illumination environment (brightness, colorfulness, and saturation). In practice, colorfulness, chroma, and saturation are all interrelated, so some have suggested that 5 dimensions may be sufficient for describing color appearance. A CAM includes a **chromatic adaptation transform (CAT)** whose goal is to achieve some measure of color constancy by adapting the white point, and it also includes equations for calculating the 6 values just mentioned.

9.5 Color Spaces

Now that we have considered the fundamentals of color sensation and representation, in this section we examine several of the more commonly used color spaces, along with the transformations between them.

9.5.1 *RGB* and *R′G′B′*

The color space most natural for cameras and displays is *RGB*. For cameras, the term *RGB* refers to the fact that the peaks in the SSFs of the filters placed over the pixels' photodetectors correspond roughly to red, green, and blue; for displays the term refers to the fact that the three primaries used look like red, green, and blue. To carefully distinguish between the values in the linear light space in which Grassmann's Law applies, and the nonlinear values resulting from gamma compression, we use *RGB* to refer to the former and *R′G′B′* to refer to the latter. If you load pixel values from a file containing a color image, you are most likely accessing nonlinear *R′G′B′*, not linear *RGB*, unless the pixel values were stored in raw format.

CamScanner

It is important to note that *RGB* values do not uniquely define a color, because the values depend upon which color space is being used. An *RGB* space is uniquely defined by its primaries and its white point (along with the transfer function used for gamma compression in the case of nonlinear $R'G'B'$). Given the linear *XYZ* tristimulus values of the three primaries and the whitepoint, the transformation from linear *RGB* to *XYZ* is

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix}}_{\text{primaries}} \begin{bmatrix} A_r & 0 & 0 \\ 0 & A_g & 0 \\ 0 & 0 & A_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{9.32}$$

where the scaling coefficients A_r, A_g, and A_b are given by solving

$$\underbrace{\begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix}}_{\text{primaries}} \begin{bmatrix} A_r \\ A_g \\ A_b \end{bmatrix} = \underbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}}_{\text{white point}} \tag{9.33}$$

In practice the overall intensities of the primaries are not available. Instead we have access only to their chromaticity coordinates (x_r, y_r), (x_g, y_g), and (x_b, y_b). Inserting $\mathbf{F}^{-1}\mathbf{F}$ just after the matrix containing the primaries in Equations (9.32) and (9.33), where $\mathbf{F} \equiv diag(Y_r, Y_g, Y_b)$, yields

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \begin{bmatrix} Y_r^{-1} & 0 & 0 \\ 0 & Y_g^{-1} & 0 \\ 0 & 0 & Y_b^{-1} \end{bmatrix} \begin{bmatrix} Y_r & 0 & 0 \\ 0 & Y_g & 0 \\ 0 & 0 & Y_b \end{bmatrix} \begin{bmatrix} A_r & 0 & 0 \\ 0 & A_g & 0 \\ 0 & 0 & A_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{9.34}$$

$$\begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \begin{bmatrix} Y_r^{-1} & 0 & 0 \\ 0 & Y_g^{-1} & 0 \\ 0 & 0 & Y_b^{-1} \end{bmatrix} \begin{bmatrix} Y_r & 0 & 0 \\ 0 & Y_g & 0 \\ 0 & 0 & Y_b \end{bmatrix} \begin{bmatrix} A_r \\ A_g \\ A_b \end{bmatrix} = \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} \tag{9.35}$$

which lead to equations that we can use:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_r/y_r & x_g/y_g & x_b/y_b \\ 1 & 1 & 1 \\ z_r/y_r & z_g/y_g & z_b/y_b \end{bmatrix} \begin{bmatrix} \alpha_r & 0 & 0 \\ 0 & \alpha_g & 0 \\ 0 & 0 & \alpha_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{9.36}$$

$$\begin{bmatrix} x_r/y_r & x_g/y_g & x_b/y_b \\ 1 & 1 & 1 \\ z_r/y_r & z_g/y_g & z_b/y_b \end{bmatrix} \begin{bmatrix} \alpha_r \\ \alpha_g \\ \alpha_b \end{bmatrix} = \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} \tag{9.37}$$

since $x_r \equiv X_r/S_r$, $y_r \equiv Y_r/S_r$, and $\alpha_r \equiv A_r Y_r$, where $S_r \equiv X_r + Y_r + Z_r$, so that $X_r/Y_r = x_r S_r / y_r S_r = x_r/y_r$, and similarly for the other color channels. If we further assume that the values for *R*, *G*, and *B* range from 0 to 1, and similarly for *Y*, and if we assume that the maximum intensity of all three primaries $(R = G = B = 1)$ leads to the maximum luminance $(Y = 1)$, then we have the additional constraint that $\alpha_r + \alpha_g + \alpha_b = 1$.

Putting this all together, if we are given the chromaticities of the primaries and the white-point, Equation (9.37) can be solved for α_r, α_g, and α_b. Once these values have been found, we then impose the constraint that $\alpha_r + \alpha_g + \alpha_b = 1$ to remove the effects of any scaling

in (X_w, Y_w, Z_w). Then these normalized values are plugged back into Equation (9.36) to yield the transformation from (R, G, B) to (X, Y, Z).

The chromaticities and white points of several well-known video and image standards are listed in Table 9.1. (Their nonlinear transfer functions were covered in a previous chapter.*) The analog color television signal broadcast in the United States and Japan, among other countries, from its adoption in 1953 to its replacement by digital video in 2009, is known as **NTSC** (for the National Television System Committee). The NTSC primaries correspond to the phosphor technology used at the time, and the illuminant is the now-obsolete CIE Standard Illuminant *C*. Phosphor technology improved considerably over the next decade, so that when the European equivalent **PAL** (for Phase Alternating Line) was standardized in 1966, the primaries were changed to something very close to those in use today. The chromaticity coordinates and white points of these NTSC and PAL standards were essentially inherited by the two versions of **Rec. 601**,† which was adopted in 1982 for encoding interlaced analog video signals as digital video.

EXAMPLE 9.1

Compute the 3×3 matrix conversion between *RGB* and CIE *XYZ* using the NTSC primaries and white point shown in Table 9.1.

Solution: From the first row of the table we have $x_r/y_r = 0.67/0.33 = 2.03$, and $z_r/y_r = (1 - x_r - y_r)/y_r = (1 - 0.67 - 0.33)/0.33 = 0$ for the red channel. Repeating for the other color channels, Equation (9.37) is used to solve for the coefficients:

$$\begin{bmatrix} \alpha_r \\ \alpha_g \\ \alpha_b \end{bmatrix} = \begin{bmatrix} 2.0303 & 0.2958 & 1.75 \\ 1 & 1 & 1 \\ 0 & 0.1127 & 9.75 \end{bmatrix}^{-1} \begin{bmatrix} 0.310 \\ 0.316 \\ 0.374 \end{bmatrix} = \begin{bmatrix} 0.0945 \\ 0.1853 \\ 0.0362 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.2990 \\ 0.5864 \\ 0.1146 \end{bmatrix} \tag{9.38}$$

where the final values are obtained by normalizing so that $\alpha_r + \alpha_g + \alpha_b = 1$. Plugging these values back into Equation (9.36) yields the desired result:

$$\begin{bmatrix} X^C \\ Y^C \\ Z^C \end{bmatrix} = \begin{bmatrix} 2.0303 & 0.2958 & 1.75 \\ 1 & 1 & 1 \\ 0 & 0.1127 & 9.75 \end{bmatrix} \begin{bmatrix} 0.2990 & 0 & 0 \\ 0 & 0.5864 & 0 \\ 0 & 0 & 0.1146 \end{bmatrix} \begin{bmatrix} R^C_{NTSC} \\ G^C_{NTSC} \\ B^C_{NTSC} \end{bmatrix} \tag{9.39}$$

$$= \underbrace{\begin{bmatrix} 0.6070 & 0.1734 & 0.2006 \\ 0.2990 & 0.5864 & 0.1146 \\ -0.0000 & 0.0661 & 1.1175 \end{bmatrix}}_{\mathbf{M}^C_{NTSC}} \begin{bmatrix} R^C_{NTSC} \\ G^C_{NTSC} \\ B^C_{NTSC} \end{bmatrix} \tag{9.40}$$

where we have introduced the superscript to indicate the white point being used (in this case CIE Standard Illuminant *C*) and the subscript to indicate the choice of *RGB* primaries (in this case those specified by NTSC), and the notation $\mathbf{M}^a_b$ indicates a transform from the primaries in *b* using the white point *a*. What is particularly interesting about this result is the middle row, which says that the luminance is given by

$$Y^C = 0.299R^C_{601} + 0.587G^C_{601} + 0.114B^C_{601} \tag{9.41}$$

where we have changed the subscript to 601 because these are the numbers adopted by Rec. 601. This equation is the basis behind the well-known **3-6-1 rule** used to convert RGB to grayscale, which we shall examine in more detail later.

* Section 2.3.2 (p. 43).

† Officially ITU-R Recommendation BT.601, formerly known as CCIR 601.

	red	green	blue	white point	
NTSC	(0.67, 0.33)	(0.21, 0.71)	(0.14, 0.08)	(0.310, 0.316)	CIE C
PAL	(0.64, 0.33)	(0.29, 0.60)	(0.15, 0.06)	(0.3127, 0.3290)	CIE D_{65}
Rec. 709, sRGB	(0.640, 0.330)	(0.300, 0.600)	(0.150, 0.060)	(0.3127, 0.3290)	CIE D_{65}
Rec. 2020	(0.708, 0.292)	(0.170, 0.797)	(0.131, 0.046)	(0.3127, 0.3290)	CIE D_{65}

TABLE 9.1 The (x,y) chromaticity coordinates of the primaries and white point of several video and image standards. NTSC and PAL are the original North American and European analog video signals, respectively, no longer in widespread use; Rec. 709 is the modern standard used in HDTV, sRGB is the standard for desktop computing, and Rec. 2020 is the latest standard used in UHDTV.

The modern video and image standards used by nearly all devices today share the same chromaticities and white points as each other, differing only in their transfer function. **Rec. 709**[†] was adopted in 1990 as the standard for high-definition television (HDTV), as well as for studio video and broadcast television. The **sRGB** standard was developed in 1996 by leading personal computer manufacturers as a standard for personal desktop computers, monitors, printers, and the Internet, and it is also used by LCD monitors, digital cameras, and scanners. **Rec. 2020** was adopted in 2012 as the standard for ultra-high-definition television (UHDTV). As a general rule one should assume, in the absence of an embedded color profile or additional information, that any 8-bit-per-channel still image is in the sRGB color space, while any video is in that of Rec. 709.

Repeating the procedure above for the Rec. 709 standard (or equivalently, sRGB, since the linear color spaces are the same) yields

$$\begin{bmatrix} X^{D65} \\ Y^{D65} \\ Z^{D65} \end{bmatrix} = \underbrace{\begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix}}_{\mathbf{M}_{709}^{D65}} \begin{bmatrix} R_{709}^{D65} \\ G_{709}^{D65} \\ B_{709}^{D65} \end{bmatrix} \tag{9.42}$$

Thus the more modern color space leads to a **2-7-1** rule for computing luminance:

$$Y^{D65} = 0.2126 R_{709}^{D65} + 0.7152 G_{709}^{D65} + 0.0722 B_{709}^{D65} \tag{9.43}$$

Note that in both Equations (9.40) and (9.42) the middle row sums to 1, and each column of the matrix contains the tristimulus values of the corresponding primary. Thus, $0.607/(0.607 + 0.299) = 0.67$, $0.1734/(0.1734 + 0.5864 + 0.0661) = 0.21$, $0.2006/(0.2006 + 0.1146 + 1.1175) = 0.14$, and so on. And of course the conversion from XYZ back to RGB is simply the inverse of the matrix.

When converting from one RGB space to another, it is common to transform the RGB values to a standard reference coordinate system (e.g., CIE XYZ), then convert from the reference coordinate system to the other RGB space. That is, the overall transform involves multiplying one 3×3 matrix by the inverse of the other matrix, but care must be taken because an additional matrix needs to be introduced if the white points are different. For example, although the matrices above transform both $NTSC$ and Rec. 709 to CIE XYZ, they use different white points, so the transform between them is actually $(\mathbf{M}_{709}^{D65})^{-1}\,\mathbf{N}_{C}^{D65}\,\mathbf{M}_{NTSC}^{C}$, where $\mathbf{N}_{b}^{a}$ transforms between the two white points a and b. This is very important because, although Rec. 709 and sRGB use D_{65}, the ICC profiles used for transforming between color spaces use D_{50} instead.

[†] Officially ITU-R Recommendation BT.709.

EXAMPLE 9.2

Compute the 3×3 matrix conversion between D_{65} and D_{50}.

Solution: Since we are free to use any primaries we like, let us use the NTSC primaries. Similar to Example 9.1, we have the following for D_{50}:

$$\begin{bmatrix} \alpha_r \\ \alpha_g \\ \alpha_b \end{bmatrix} = \begin{bmatrix} 2.0303 & 0.2958 & 1.75 \\ 1 & 1 & 1 \\ 0 & 0.1127 & 9.75 \end{bmatrix}^{-1} \begin{bmatrix} 0.3457 \\ 0.3585 \\ 0.2958 \end{bmatrix} = \begin{bmatrix} 0.1148 \\ 0.2158 \\ 0.0278 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.3203 \\ 0.6020 \\ 0.0777 \end{bmatrix} \tag{9.44}$$

so that

$$\begin{bmatrix} X^{D50} \\ Y^{D50} \\ Z^{D50} \end{bmatrix} = \begin{bmatrix} 2.0303 & 0.2958 & 1.75 \\ 1 & 1 & 1 \\ 0 & 0.1127 & 9.75 \end{bmatrix} \begin{bmatrix} 0.3203 & 0 & 0 \\ 0 & 0.6020 & 0 \\ 0 & 0 & 0.0777 \end{bmatrix} \begin{bmatrix} R_{NTSC}^{D50} \\ G_{NTSC}^{D50} \\ B_{NTSC}^{D50} \end{bmatrix} \tag{9.45}$$

$$= \underbrace{\begin{bmatrix} 0.6503 & 0.1781 & 0.1359 \\ 0.3203 & 0.6020 & 0.0777 \\ -0.0000 & 0.0678 & 0.7573 \end{bmatrix}}_{\mathbf{M}_{NTSC}^{D50}} \begin{bmatrix} R_{NTSC}^{D50} \\ G_{NTSC}^{D50} \\ B_{NTSC}^{D50} \end{bmatrix} \tag{9.46}$$

Similarly, for D_{65} we have

$$\begin{bmatrix} \alpha_r \\ \alpha_g \\ \alpha_b \end{bmatrix} = \begin{bmatrix} 2.0303 & 0.2958 & 1.75 \\ 1 & 1 & 1 \\ 0 & 0.1127 & 9.75 \end{bmatrix}^{-1} \begin{bmatrix} 0.3127 \\ 0.3290 \\ 0.3583 \end{bmatrix} = \begin{bmatrix} 0.0953 \\ 0.1993 \\ 0.0344 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.2897 \\ 0.6056 \\ 0.1047 \end{bmatrix} \tag{9.47}$$

so that

$$\begin{bmatrix} X^{D65} \\ Y^{D65} \\ Z^{D65} \end{bmatrix} = \begin{bmatrix} 2.0303 & 0.2958 & 1.75 \\ 1 & 1 & 1 \\ 0 & 0.1127 & 9.75 \end{bmatrix} \begin{bmatrix} 0.2897 & 0 & 0 \\ 0 & 0.6056 & 0 \\ 0 & 0 & 0.1047 \end{bmatrix} \begin{bmatrix} R_{NTSC}^{D65} \\ G_{NTSC}^{D65} \\ B_{NTSC}^{D65} \end{bmatrix} \tag{9.48}$$

$$= \underbrace{\begin{bmatrix} 0.5881 & 0.1791 & 0.1832 \\ 0.2897 & 0.6056 & 0.1047 \\ -0.0000 & 0.0682 & 1.0208 \end{bmatrix}}_{\mathbf{M}_{NTSC}^{D65}} \begin{bmatrix} R_{NTSC}^{D65} \\ G_{NTSC}^{D65} \\ B_{NTSC}^{D65} \end{bmatrix} \tag{9.49}$$

The conversion from D_{65} to D_{50} is then given by

$$\begin{bmatrix} X^{D50} \\ Y^{D50} \\ Z^{D50} \end{bmatrix} = \underbrace{\mathbf{M}_{NTSC}^{D50}(\mathbf{M}_{NTSC}^{D65})^{-1}}_{\mathbf{N}_{D65}^{D50}} \begin{bmatrix} X^{D65} \\ Y^{D65} \\ Z^{D65} \end{bmatrix} = \underbrace{\begin{bmatrix} 1.1206 & -0.0301 & -0.0649 \\ 0.0621 & 0.9797 & -0.0355 \\ -0.0164 & 0.0333 & 0.7414 \end{bmatrix}}_{\mathbf{N}_{D65}^{D50}} \begin{bmatrix} X^{D65} \\ Y^{D65} \\ Z^{D65} \end{bmatrix} \tag{9.50}$$

9.5.2 $Y'P_BP_R$

It is often desirable to separate the **luminance**, which is related to the perception of brightness, from the **chrominance**, which captures the parameters that we normally associate with the notion of color (e.g., hue and chroma). This separation is performed by transforming the three *RGB* sensor values into a color space in which the luminance is described by one axis, while the chrominance is described by the remaining two axes. As we shall see, nearly all color spaces except for RGB / $R'G'B'$ are based upon such a separation. (The *XYZ* space, while not explicitly maintaining this separation, can nevertheless be transformed by simply concatenating the luminance value Y with the chromaticity coordinates x and y to yield the xyY color space.)

The first such color space that we shall consider is $Y'P_BP_R$, which is sometimes used for transmitting analog video signals. The prime (′) indicates that the value Y' is not the luminance used in color science. In fact, it is not even gamma-corrected luminance but rather the nonlinear quantity known to video engineers as **luma**, which is computed as the weighted average of the three gamma-corrected $R'G'B'$ values: $Y' = \alpha_r R' + \alpha_g G' + \alpha_b B'$. For Rec. 601 and Rec. 709, this leads to the following definitions:

$$^{601}Y' \equiv 0.299R' + 0.587G' + 0.114B' \tag{9.51}$$

$$^{709}Y' \equiv 0.2126R' + 0.7152G' + 0.0722B' \tag{9.52}$$

where the leading superscript indicates the coefficients used in computing luma, and the trailing super/subscripts have been dropped because the exact choice of primaries and white point is not important.

Compared with the middle row of the colorimetric expressions in Equations (9.41) and (9.43), the most important difference is that Equations (9.51)–(9.52) above operate in the gamma-corrected $R'G'B'$ space and, as a result, they are no longer correct in any colorimetric sense. To understand the reason behind defining luma in a non-colorimetric way, we must consider the historical context in which the decision was made. As shown in Figure 9.12, the correct way of transmitting *RGB* values back in the days of analog video transmission would have been, in the encoder, to convert to luminance Y (and chrominance) using a 3×3 transform, then apply gamma compression to reduce the perceptual effects of noise by transmitting the luma Y' instead of the luminance Y. The decoder would then have applied gamma expansion to recover the luminance, then applied the inverse 3×3 transform to yield linear *RGB* values. So far so good. However, because cathode ray tube (CRT) displays, which were the prevailing technology in the 1950s, inherently apply a nonlinear transform function to the input voltages, a nonlinear function resembling gamma compression had to be inserted to cancel this effect. Therefore, to simplify the decoding process, video engineers reversed the order of the gamma and linear transform blocks, thus rendering two of the blocks in the diagram unnecessary since they cancel each other. This engineering tradeoff sacrificed colorimetric accuracy for implementation simplicity. Although this tradeoff would not be necessary with today's technology, the effects of this decision are still with us due to their influence upon widely-used video standards.

The nonlinear chrominance components,[1] P_B and P_R, are the color differences computed with respect to the blue and red channels, since green is the most similar to luma:

$$P_B \equiv \frac{1}{2} \cdot \frac{B' - Y'}{1 - \alpha_b} = \frac{-\alpha_r R' - \alpha_g G' + (1 - \alpha_b)B'}{2(1 - \alpha_b)} \tag{9.53}$$

[1] Just as nonlinear luminance is called *luma*, so nonlinear chrominance is called *chroma*, Section 2.3.4 (p. 52). However, to avoid confusion with the other meaning of the term "chroma", we will avoid the use of the term here.

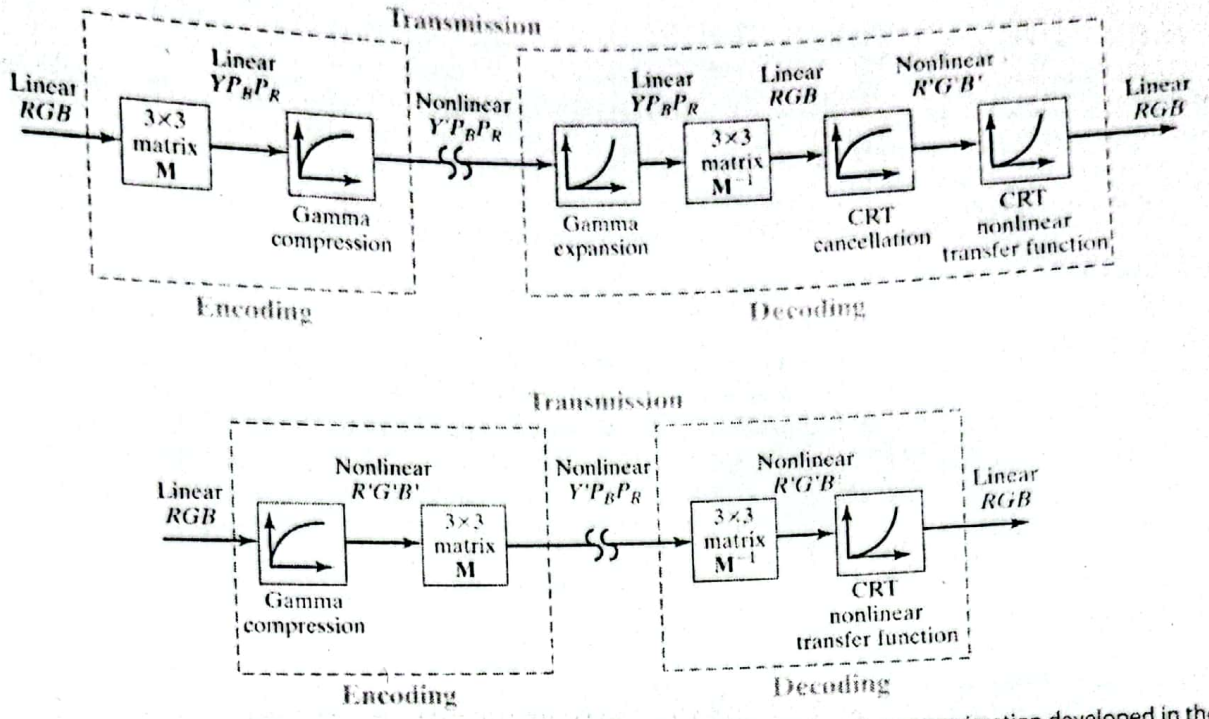


Figure 9.12 The theoretically correct way of transmitting analog *RGB* (top), and the engineering approximation developed in the 1950s (bottom). The approximation swaps the order of the linear 3 × 3 transform and the nonlinear gamma compression or expansion, to simplify decoding by canceling two of the blocks in the diagram.

$$P_R = \frac{1}{2} \cdot \frac{R' - Y'}{1 - \alpha_r} = \frac{(1 - \alpha_r)R' - \alpha_g G' - \alpha_b B'}{2(1 - \alpha_r)} \tag{9.54}$$

The scaling factors ensure that P_B and P_R range from −0.5 to +0.5, whereas Y' ranges from 0 to 1. Plugging in the Rec. 601 coefficients leads to the transformation from $R'G'B'$ to ${}^{601}Y'P_BP_R$:

$$\begin{bmatrix} {}^{601}Y' \\ P_B \\ P_R \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/(2\cdot 0.886) & 0 \\ 0 & 0 & 1/(2\cdot 0.701) \end{bmatrix} \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \tag{9.55}$$

$$= \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \tag{9.56}$$

and similarly using the Rec. 709 coefficients:

$$\begin{bmatrix} {}^{709}Y' \\ P_B \\ P_R \end{bmatrix} = \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1146 & -0.3854 & 0.5 \\ 0.5 & -0.4542 & -0.0458 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \tag{9.57}$$

Note that in both cases the first row sums to 1, while the second and third rows sum to zero. Conversion back to $R'G'B'$ is obtained by simply inverting the matrix in each case.

9.5.3 $Y'C_BC_R$

More common than $Y'P_BP_R$, which is used for analog signals, is $Y'C_BC_R$, which is used for digital signals. The latter is a scaled and offset version of the former:

$$\begin{bmatrix} {}_{219}Y' \\ C_B \\ C_R \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 219Y' \\ 224P_B \\ 224P_R \end{bmatrix} \tag{9.58}$$

where the leading subscript 219 indicates the scaling of luma. Note that ${}_{219}Y'$, C_B, and C_R on the left side each fit within 8 bits: $16 \leq {}_{219}Y' \leq 235$ and $16 \leq C_B, C_R \leq 240$. At first it may seem strange that the full range of 256 values is not used, but this extra *footroom* (below 16) and *headroom* (above 235 for luma or 240 for nonlinear chrominance) is necessary to allow for some undershoot or overshoot when filtering, without incurring clipping. Combining Equations (9.56) and (9.58) yields the transform from 8-bit $R'G'B'$ to 8-bit ${}^{601}_{219}Y'C_BC_R$ using the Rec. 601 coefficients:

$$\begin{bmatrix} {}^{601}_{219}Y' \\ C_B \\ C_R \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \frac{1}{255} \begin{bmatrix} 65.4810 & 128.5530 & 24.9660 \\ -37.7968 & -74.2032 & 112.0000 \\ 112.0000 & -93.7860 & -18.2140 \end{bmatrix} \begin{bmatrix} {}_{255}R' \\ {}_{255}G' \\ {}_{255}B' \end{bmatrix} \tag{9.59}$$

where we have added a scaling factor of 1/255 so that the equation applies when ${}_{255}R'$, ${}_{255}G'$, and ${}_{255}B'$ are also stored as 8 bits each, i.e., $0 \leq {}_{255}R', {}_{255}G', {}_{255}B' \leq 255$.

The JFIF format for JPEG uses Rec. 601 and no headroom or footroom, so that Y', C_B, and C_R encompass the full range from 0 to 255. With R', G', and B' also ranging from 0 to 255, the conversion is given by adding offsets to Equation (9.56):

$$\begin{bmatrix} {}^{601}_{255}Y' \\ C_B \\ C_R \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{bmatrix} \begin{bmatrix} {}_{255}R' \\ {}_{255}G' \\ {}_{255}B' \end{bmatrix} \tag{9.60}$$

Conversion back to $R'G'B'$ is obtained by simply inverting these equations. But when headroom and footroom exist, care must be taken in the inversion because if any processing of the image has occurred, some values might be in the invalid range below 16 or above 235 (in the case of Y') or above 240 (in the case of C_B or C_R). Clamping such values can lead to undesirable visual distortions in the result.

As with $Y'P_BP_R$, $Y'C_BC_R$ is not an absolute color space but rather a way of encoding *RGB* information. The actual color displayed depends on the *RGB* primaries used, the white point, and the nonlinear transfer function. A value expressed as $Y'C_BC_R$ is only predictable if these quantities are known, or if an ICC profile is used.

9.5.4 $Y'UV$

Originally the term $Y'UV$ referred to the color space used by the transmission of PAL analog video signals. The original scaling factors set the maximum U and V values to 0.436 and 0.615, respectively, instead of 0.5. From Equations (9.53) and (9.54) this yields

$$^{601}Y' \equiv \alpha_r R' + \alpha_g G' + \alpha_b B' = 0.299R' + 0.587G' + 0.114B \tag{9.61}$$

$$U \equiv 0.436 \cdot \frac{B' - Y'}{1 - \alpha_b} = 0.4921(B' - Y') \tag{9.62}$$

$$V \equiv 0.615 \cdot \frac{R' - Y'}{1 - \alpha_r} = 0.8773(R' - Y') \tag{9.63}$$

CamScanner

Surprisingly, PAL uses the Rec. 601 coefficients to convert to luma, as in Equation (9.51), even though the PAL primaries are different from those of NTSC.

Since the advent of digital video, the term $Y'UV$ has adopted a more ambiguous meaning and should generally be avoided in favor of the more specific $Y'C_BC_R$. Often, $Y'UV$ simply means some scaled version of $Y'C_BC_R$, and it is essential to know the scaling factors in order to properly interpret the values. One commonly used definition is the headroom/footroom version of Equation (9.59), which can be approximated by integer-only computations as:

$$\begin{bmatrix} {}^{601}_{219}Y' \\ U \\ V \end{bmatrix} \approx \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \frac{1}{256}\begin{bmatrix} 66 & 129 & 25 \\ -38 & -74 & 112 \\ 112 & -94 & -18 \end{bmatrix}\begin{bmatrix} {}_{255}R' \\ {}_{255}G' \\ {}_{255}B' \end{bmatrix} \tag{9.64}$$

with the inverse transform given by

$$\begin{bmatrix} {}_{255}R' \\ {}_{255}G' \\ {}_{255}B' \end{bmatrix} \approx \frac{1}{256}\begin{bmatrix} 298 & 0 & 409 \\ 298 & -100 & -208 \\ 298 & 516 & 0 \end{bmatrix}\left(\begin{bmatrix} {}^{601}_{219}Y' \\ U \\ V \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}\right) \tag{9.65}$$

9.5.5 $Y'IQ$

As $Y'UV$ was the color space used by analog PAL, the color space used originally by analog NTSC was $Y'IQ$, which is $Y'UV$ with the chrominance axes rotated by 33 degrees and exchanged:

$$\begin{bmatrix} {}^{601}Y' \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\sin(33°) & \cos(33°) \\ 0 & \cos(33°) & \sin(33°) \end{bmatrix}\begin{bmatrix} {}^{601}Y' \\ U \\ V \end{bmatrix} \tag{9.66}$$

However, even NTSC switched from using $Y'IQ$ to $Y'UV$ in the early 1970s, so that $Y'IQ$ has been obsolete for a long time now.

9.5.6 Converting from $R'G'B'$ to Grayscale

We are now in a position to consider a problem that arises frequently in practice, namely, how to properly convert an $R'G'B'$ image to grayscale. Conventional wisdom suggests that one should always use the 3-6-1 rule of Equation (9.51):

$$gray = 0.299R' + 0.587G' + 0.114B' \tag{9.67}$$

This is the transformation you will most often find in image processing and computer vision software. However, as we saw in our analysis above, this transformation has never been correct in a colorimetric sense because it applies coefficients derived from linear RGB space to a nonlinear $R'G'B'$ space; moreover, it is based upon obsolete primary chromaticities and an obsolete white point. In contrast, the proper way to convert $R'G'B'$ to grayscale is to apply the appropriate inverse nonlinear transfer function (gamma expansion) to yield linear RGB values, then apply the Rec. 709 coefficients of Equation (9.52), then apply the appropriate nonlinear transfer function (gamma compression). An easy way to convince yourself of the potential error that can occur by ignoring gamma is to apply Equation (9.67) to a purely blue image ($R' = G' = 0$, $B' = 255$ for all pixels); the result will be $gray = 29$ everywhere, which will be noticeably darker than the original.

Having said that, it is comforting to know that, unless the particular application at hand requires colorimetric fidelity, there is no need to go to all this trouble. One of the reasons

the 3-6-1 rule is still widely used is that it works well in practice, but in fact just about any reasonably weighted average of the three components yields a visually acceptable result, as we saw earlier.[1] For many applications, therefore, simpler methods of conversion are feasible, such as the one we introduced in Equation (3.37),

$$gray = \frac{1}{4}(R' + 2G' + B') \tag{9.68}$$

which has the advantage that it can be implemented with only bitshifts and without any floating-point multiplication or division. As an aside, note that (as we discussed previously) the inverse conversion from grayscale to $R'G'B'$ is easy: simply replicate the value three times.

9.5.7 Opponent Colors

Because of their effective use by the human visual system, opponent color spaces are often useful for image processing applications. One natural approach is to define the two color dimensions c_1 and c_2 as blue minus yellow and red minus green, respectively, as in the human visual system:

$$c_1 \equiv \text{blue} - \text{yellow} = B' - \frac{1}{2}(R' + G') \tag{9.69}$$

$$c_2 \equiv \text{red} - \text{green} = R' - G' \tag{9.70}$$

leading to

$$\begin{bmatrix} Y' \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} \alpha_r & \alpha_g & \alpha_b \\ -\frac{1}{2} & -\frac{1}{2} & 1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \tag{9.71}$$

where Y' is the luma, and we can set $\alpha_r = \alpha_g = \alpha_b = \frac{1}{3}$ for simplicity or use one of the standard sets of luma coefficients.

An alternative, which we shall call the **Hanbury transformation**, is to define c_1 and c_2 as follows:

$$\begin{bmatrix} Y' \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} \alpha_r & \alpha_g & \alpha_b \\ 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \tag{9.72}$$

which has the advantage that it maps the corners of the $R'G'B'$ color cube to a hexagon in the c_1-c_2 chromaticity plane, a concept explored in more detail next.

9.5.8 *HSV* and *HSL*

Consider the $R'G'B'$ color cube shown in Figure 9.13, whose eight vertices correspond to the three additive and subtractive primaries (red, green, blue, cyan, magenta, and yellow), along with black and white. By convention black is at (0, 0, 0), white is at (1, 1, 1), and red, green, and blue are at (1, 0, 0), (0, 1, 0), and (0, 0, 1), respectively. Note that the R', G', and B' axes form a righthand coordinate system. Also shown in the figure is the line of grays connecting the white and black vertices, along with an arbitrary point at $\mathbf{p} = (0.8, 0.6, 0.4)$

[1] Section 3.4.1 (p. 93).

for illustrative purposes. While one could also define the RGB color cube using linear light intensities, it rarely (if ever) makes sense to do so, since it is usually desirable to preserve perceptual uniformity; hence the choice of $R'G'B'$ here.

A curious property of the $R'G'B'$ color cube is that, for each additive primary (red, green, or blue), there is exactly one face that connects it with three other vertices, each of which is not an additive primary. Thus, the red face connects red with magenta, yellow, and white; the green face connects green with cyan, yellow, and white; and the blue face connects blue with cyan, magenta, and white. *If the cube is rotated by 45 degrees about the R' axis, then rotated again by $\hat{\theta}$ about the original G' axis, where $\hat{\theta} \approx -35.26$ degrees,*[†] then the cube will be balanced on the black vertex, with the white vertex directly above it. Viewed from above with orthographic projection, the cube adopts the shape of a regular hexagon whose six vertices are the three additive and three subtractive primaries, with the gray axis coming out of the page and piercing the center of the hexagon. Compared with Figure 9.3, we see that this hexagon is simply our old friend the color wheel, with a slightly different shape. Mathematically the transformation from $R'G'B'$ to color difference axes c_1 and c_2 is obtained by concatenating the rotation matrices, along with appropriate scaling factors for the nonlinear chromaticity and luma axes:

$$\begin{bmatrix} c_1 \\ c_2 \\ Y' \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{3}{2}} & 0 & 0 \\ 0 & \sqrt{\frac{3}{2}} & 0 \\ 0 & 0 & \frac{1}{3}\sqrt{3} \end{bmatrix} \begin{bmatrix} \cos\hat{\theta} & 0 & \sin\hat{\theta} \\ 0 & 1 & 0 \\ -\sin\hat{\theta} & 0 & \cos\hat{\theta} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 45^\circ & -\sin 45^\circ \\ 0 & \sin 45^\circ & \cos 45^\circ \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \tag{9.73}$$

$$= \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \tag{9.74}$$

which is simply the Hanbury transformation with $\alpha_r = \alpha_g = \alpha_b = \frac{1}{3}$, and a reordering of the rows. Notice that this transformation maps red $(R' = 1, G' = B' = 0)$ to $(c_1, c_2) = (1, 0)$, green $(G' = 1, R' = B' = 0)$ to $(-\frac{1}{2}, \frac{\sqrt{3}}{2})$, and blue $(B' = 1, R' = G' = 0)$ to $(-\frac{1}{2}, -\frac{\sqrt{3}}{2})$.

The hexagon is naturally divided into six equally-sized sectors, and points fall into one or the other sector depending upon the relative values of R', G', and B'. By convention the hue is defined as the counterclockwise angle with respect to the red axis, so that sector 0 contains all the points such that $R' \geq G' \geq B'$ (hue from 0 to 60 degrees), sector 1 contains all the points such that $G' \geq R' \geq B'$ (hue from 60 to 120 degrees), and so forth. R' is the maximum value in sectors 0 and 5, G' is the maximum value in sectors 1 and 2, and B' is the maximum value in sectors 3 and 4. These relationships can be seen from the colored faces of the 3D color cube, or from the lines that bisect the primary colors in the 2D hexagon.

For any given color (R', G', B') located at a certain point within the cube, the point lies within a **subcube** anchored at black whose length along each side is $V_{max} = \max(R', G', B')$, shown in the left side of Figure 9.13. This subcube projects orthographically onto the page as a smaller hexagon within the complete hexagon already mentioned. For clarity, we shall refer to the smaller hexagon as a **sub-hexagon**. The size of the sub-hexagon depends on V_{max}, with $V_{max} = 1$ yielding the complete hexagon, and $V_{max} = 0$ yielding a point. If we stack all these sub-hexagons on top of one another we get a hexagon-based pyramid called a **hexcone**, which is illustrated in Figure 9.14.

With these preliminaries in place, we are now in a position to begin to quantify the concepts of hue, chroma, saturation, and value that we mentioned earlier in the continuum

[†] The exact angle is given by $\cos\hat{\theta} = \sqrt{2/3}$, $\sin\hat{\theta} = -1/\sqrt{3}$.

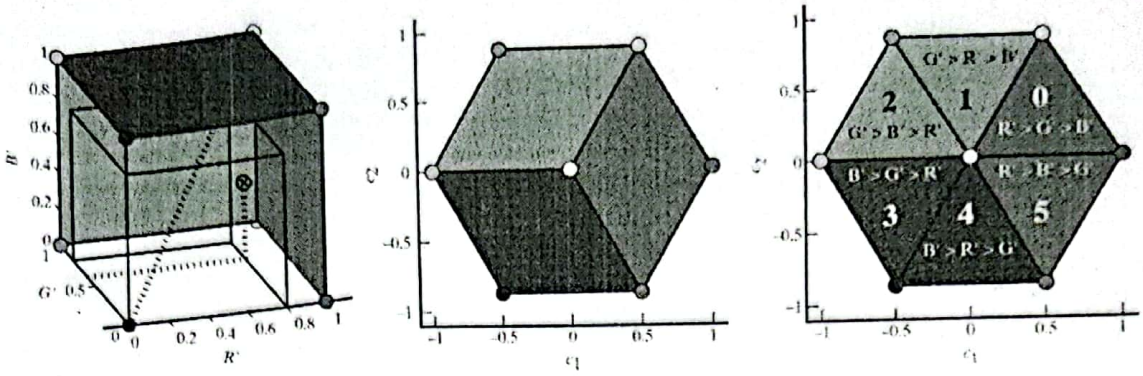
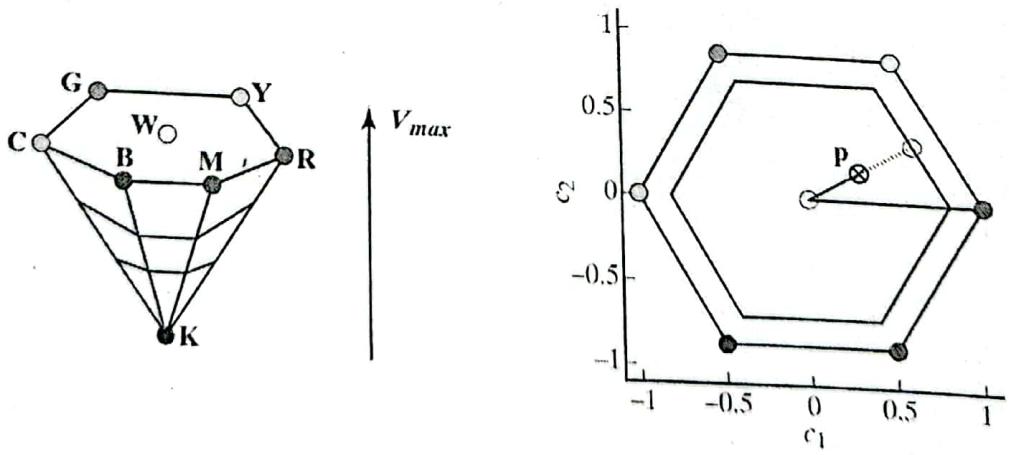


Figure 9.13 LEFT: The $R'G'B'$ color cube, with the red, green, and blue faces colored. Also shown is the line of grays and an arbitrary point at $\mathbf{p} = (0.8, 0.6, 0.4)$ along with its subcube. Middle: Top-down view of the cube after rotating so that white is above black. RIGHT: The 2D hexagon within which all (R', G', B') colors project orthographically parallel to the line of grays. The hexagon is divided into six sectors (numbered 0 through 5), and each point falls into a particular sector depending on the relative ordering of R', G', and B'.

the Munsell color system. The *HSV* and *HSL* color spaces were developed in the 1970s for computer graphics applications. Specifically, they are used widely in "color picker" user interfaces that allow a person to select a color in a more intuitive way than by specifying $R'G'B'$ values directly. They are also commonly used in image processing and computer vision applications because they separate luma from nonlinear chrominance, placing the latter values along axes that facilitate, to some extent, meaningful color differences.

A confusing set of choices exists in defining a color space based on these principles, which over time has led to a mishmash of related techniques all going by different but similar-sounding names. Thus, we have *HSV* (hue-saturation-value), *HSL* (hue-saturation-lightness), *HSI* (hue-saturation-intensity), *HSB* (hue-saturation-brightness), and so forth. In many of these cases the names are misleading because the terms are not used precisely, and the

Figure 9.14 LEFT: The hexcone obtained by stacking the sub-hexagons associated with each value of V_{max}. RIGHT: The point $\mathbf{p} = (0.8, 0.6, 0.4)$ projects onto the nonlinear chromaticity plane at $(c_1, c_2) \approx (0.3, 0.2)$, which lies within sector 0. Its sub-hexagon with spokes of length 0.8 is shown within the complete hexagon, and the intersection of the ray toward $\mathbf{p}$ with the sub-hexagon is shown at $\approx (0.6, 0.35)$. For any point, the hue is the counterclockwise angle with respect to the red axis, the chroma is the distance from the center to the point, and the saturation is the chroma normalized by the distance from the center to the sub-hexagon along the ray passing through $\mathbf{p}$.



differences between the spaces are not necessarily captured by the names. For example, the term "value" in *HSV* is not used in the standard way, *HSL* does not use true "lightness", *HSI* usually does not denote true linear "intensity", and *HSB* does not capture actual "brightness", which is a subjective concept. Similarly, the "hue" in *HSV* slightly differs from that of *HSL*.

To avoid such confusion, we shall describe the original formulations of *HSV* and *HSL*, treating these as the official definitions, being careful to maintain the proper distinction between the two. In truth, they are both quite similar, with the most significant difference being that *HSV* is based on orthographic projection, whereas *HSL* is based on perspective projection. This choice primarily affects the saturation, which, as we will show, is not really the best quantity to be measuring anyway.

HSV

The *HSV* color space is based on the **hexcone model**, which involves orthographic projection along a direction parallel to the line of grays. The value V is simply the height within the cone, namely, V_{max}, while the saturation and hue are defined within the sub-hexagon specified by V_{max}. The saturation S is the distance from the center to the point, relative to the distance to the outer edge of the sub-hexagon along the same ray. That is, $S = 0$ is a shade of gray, while $S = 1$ means that the color is on the edge of the sub-hexagon and therefore is purely saturated. If we define $V_{min} \equiv \min(R', G', B')$, then any color can be split into the sum of a term involving two colors and a term representing a shade of gray that dilutes the color. For example, if $R' \geq G' \geq B'$, then we have $(R', G', B') = (V_\delta, G' - V_{min}, 0) + (V_{min}, V_{min}, V_{min})$, where $V_\delta \equiv V_{max} - V_{min}$. The saturation is then given by V_δ / V_{max}. The hue is approximated by H, the proportional length within the given sector along the direction parallel to the outer rim of the hexagon, added to the sector number. This definition yields an approximate hue between 0 and 6, which can then be converted to the range 0 to 1 or 0° and 360°, as desired, by a simple conversion factor.

Conversion from $R'G'B'$ to *HSV* involves computing the corresponding coordinates within the hexcone given a point in the color cube, then normalizing to compute saturation. Assuming $0 \leq R', G', B' \leq 1$, the conversion is given by

$$V \equiv V_{max} \tag{9.75}$$

$$S \equiv V_\delta / V_{max} \tag{9.76}$$

$$H \equiv 60^\circ \cdot \begin{cases} (G' - B')/V_\delta + 0 & \text{if } R' = V_{max} \quad \text{(red sectors 0 and 5)} \\ (B' - R')/V_\delta + 2 & \text{if } G' = V_{max} \quad \text{(green sectors 1 and 2)} \\ (R' - G')/V_\delta + 4 & \text{if } B' = V_{max}, \quad \text{(blue sectors 3 and 4)} \end{cases} \tag{9.77}$$

where $0 \leq S, V \leq 1$. In the definition of H, the fractions range from -1 to 1, so that when added to the central values of 0, 2, and 4 they yield a result that ranges from -1 to 5. Therefore, if the pixel is in sector 5, the resulting angle will be between -60° and 0°, so to ensure that $0^\circ \leq H < 360^\circ$, simply add 360° to the result above if $H < 0^\circ$. Note that if $S = 0$, hue is not defined due to divide by zero.

If we define $\varsigma \equiv H/60^\circ$, then the inverse transformation back to $R'G'B'$ is given by

$$(R', G', B') = \begin{cases} (V, V_3, V_1) & \text{if } \lfloor \varsigma \rfloor = 0 \quad \text{(sector 0)} \\ (V_2, V, V_1) & \text{if } \lfloor \varsigma \rfloor = 1 \quad \text{(sector 1)} \\ (V_1, V, V_3) & \text{if } \lfloor \varsigma \rfloor = 2 \quad \text{(sector 2)} \\ (V_1, V_2, V) & \text{if } \lfloor \varsigma \rfloor = 3 \quad \text{(sector 3)} \\ (V_3, V_1, V) & \text{if } \lfloor \varsigma \rfloor = 4 \quad \text{(sector 4)} \\ (V, V_1, V_2) & \text{if } \lfloor \varsigma \rfloor = 5, \quad \text{(sector 5)} \end{cases} \tag{9.78}$$

CamScanner

where

$$V_1 \equiv V(1 - S) \tag{9.79}$$

$$V_2 \equiv V(1 - \alpha S) \tag{9.80}$$

$$V_3 \equiv V(1 - (1 - \alpha)S) \tag{9.81}$$

and $\alpha \equiv s - \lfloor s \rfloor$.

The saturation $S = V_\delta / V_{max}$ can be interpreted geometrically as the chroma divided by the maximum chroma, where the chroma is defined as the distance from the hexagon center to the point, calculated as

$$C \equiv \sqrt{c_1^2 + c_2^2} = \sqrt{R'^2 + G'^2 + B'^2 - R'G' - R'B' - G'B'} \tag{9.82}$$

and the maximum chroma for the particular value V_{max} and the approximate hue H is given by the distance from the hexagon center to the sub-hexagon along the ray passing through the point, which can be shown to be†

$$C_{max} = CV_{max}/V_\delta \tag{9.83}$$

leading to saturation as the normalized chroma: $S = C/C_{max} = V_\delta / V_{max}$, which is the same as Equation (9.76).

If the exact hue angle within the hexagon is desired, it is given by

$$\tan \overline{H} = \frac{c_2}{c_1} = \frac{\sqrt{3}(G' - B')}{2R' - G' - B'} \tag{9.84}$$

where ATAN2 should be used to ensure that the resulting angle is in the correct quadrant,‡ and the overbar indicates that this is not always identical to H, although they do happen to agree whenever the hue is a multiple of 30°. Over the entire $R'G'B'$ cube, the maximum discrepancy between H in Equation (9.77) and $\overline{H}$ in Equation (9.84) is less than 1.2°.

HSL

The *HSL* color space is based on the **triangle model**, in which the point (R', G', B') is perspectively projected onto the $R' + G' + B' = 1$ plane which passes through the red, green, and blue vertices of the color cube. This projection yields the intersection of the plane with the line passing through the black vertex and the point, as shown in Figure 9.15. Given any point in the cube, its projection onto this plane lands within the triangle defined by the red, green, and blue vertices. The line of grays intersects the triangle at its centroid. The hue is the counterclockwise angle with respect to the red axis, and the saturation is the distance from the centroid to the perspectively projected point normalized by the distance from the centroid to the edge of the triangle along the ray passing through the perspectively projected point. Assuming $0 \le R', G', B' \le 1$, the conversion from $R'G'B'$ to *HSL* is given by

$$\overline{L} \equiv \frac{1}{3}(R' + G' + B') \tag{9.85}$$

$$\overline{S} \equiv 1 - \frac{V_{min}}{\overline{L}} \tag{9.86}$$

$$\overline{H} \equiv \tan^{-1} \frac{\sqrt{3}(G' - B')}{2R' - G' - B'} \tag{9.87}$$

† The rather tedious derivation is omitted.

‡ Section 4.4.4 (p. 179).

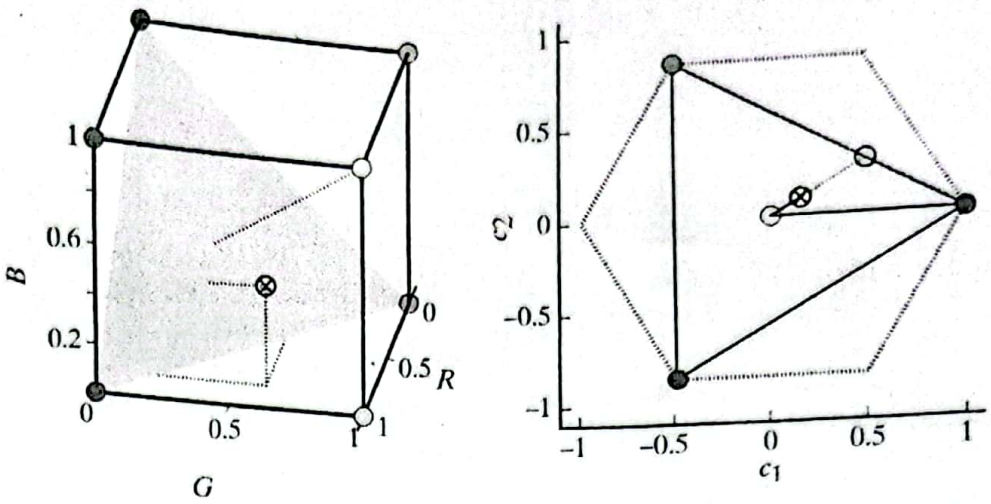


Figure 9.15 LEFT: In the *HSL* color space, a point in the $R'G'B'$ color cube is perspectively projected onto the plane defined by $R' + G' + B' = 1$, with all projections passing through the origin. RIGHT: Top-down view (white is directly above black) of the triangle defined by the intersection of the plane and the cube. The complete hexagon is shown for reference.

where the overbars are used for clarity. $\overline{H}$ is the exact angle, identical to Equation (9.84); $\overline{S}$ is the saturation in the triangle model, which is quite different from the saturation in the hexcone model; and the overbar in $\overline{L}$ reminds us that this is neither true lightness nor luminance but rather luma with coefficients $\alpha_r = \alpha_g = \alpha_b = \frac{1}{3}$. Conversion back to $R'G'B'$ depends on the sector:

$$(R', G', B') = \begin{cases} (\overline{V}_2, \overline{V}_3, V_{min}) & \text{if } 0° \le \overline{H} < 120° \quad \text{(sectors 0 and 1)} \\ (V_{min}, \overline{V}_2, \overline{V}_3) & \text{if } 120° \le \overline{H} < 240° \quad \text{(sectors 2 and 3)} \\ (\overline{V}_3, V_{min}, \overline{V}_2) & \text{if } 240° \le \overline{H} < 360° \quad \text{(sectors 4 and 5)} \end{cases} \tag{9.88}$$

where

$$V_{min} = \overline{L}(1 - \overline{S}) \tag{9.89}$$

$$\overline{V}_2 \equiv \overline{L}\left(1 + \overline{S}\frac{\cos(\overline{H} - h)}{\cos(60° + h - \overline{H})}\right) \tag{9.90}$$

$$\overline{V}_3 \equiv 3\overline{L} - (V_{min} + \overline{V}_2) \tag{9.91}$$

and $h \equiv 120° \cdot \lfloor \overline{H}/120° \rfloor$.

To understand the definition of saturation, note that the point (R', G', B') perspectively projects onto the plane at $\frac{1}{D}(R', G', B')$, where $D \equiv R' + G' + B' = 3\overline{L}$, so that $(R' + G' + B')/D = 1$. If we let (c_1, c_2) refer to the coordinates of the projection of (R', G', B') onto the c_1-c_2 plane by the Hanbury transformation in Equation (9.74), and $(\overline{c}_1, \overline{c}_2)$ be the coordinates of the projection of $\frac{1}{D}(R', G', B')$, then it is easy to see that

$$\begin{bmatrix} \overline{c}_1 \\ \overline{c}_2 \end{bmatrix} = \frac{1}{D}\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{9.92}$$

which reveals that the two points are related to each other by a simple scaling along the ray from the origin, so that both points share the same hue, that is, $\tan^{-1}\frac{c_1}{c_2} = \tan^{-1}\frac{\overline{c}_1}{\overline{c}_2}$. The chroma resulting from this perspective projection is

$$\overline{C} \equiv \sqrt{\overline{c}_1^2 + \overline{c}_2^2} = \frac{1}{D}\sqrt{c_1^2 + c_2^2} = \frac{C}{D} \tag{9.93}$$

where C is the chroma from orthographic projection defined in Equation (9.82).

As before, saturation is the normalized chroma: $\overline{S} = \frac{\overline{C}}{\overline{C}_{max}}$, where $\overline{C}_{max}$ is the distance from the triangle centroid to the intersection of the ray with the triangle, calculated as[4]

$$\overline{C}_{max} = \frac{C}{3(\overline{L} - V_{min})} \tag{9.94}$$

which leads to $\overline{S} = \frac{3(\overline{L} - V_{min})}{3\overline{L}} = 1 - \frac{V_{min}}{\overline{L}}$, as in Equation (9.86).

Sometimes you will see the following definition for hue:

$$\overline{H} = \begin{cases} \theta & \text{if } B' \leq G' \\ 360^\circ - \theta & \text{otherwise} \end{cases} \tag{9.95}$$

where

$$\theta = \cos^{-1}\left(\frac{\frac{1}{2}[(R' - G') + (R' - B')]}{\sqrt{(R' - G')^2 + (R' - B')(G' - B')}}\right) \tag{9.96}$$

or its equivalent using the trigonometry identity $\cos^{-1}\theta = 90^\circ - \tan^{-1}(\theta/\sqrt{1 - \theta^2})$. Simple algebraic manipulation reveals that both Equation (9.95) and its inverse tangent equivalent are identical to the much more compact Equation (9.87).

For completeness we should mention that the original formulation of *HSL* allows arbitrary coefficients α_r, α_g, and α_b in Equation (9.85), as long as $\alpha_r + \alpha_g + \alpha_b = 1$. However, relaxing the constraint that $\alpha_r = \alpha_g = \alpha_b = \frac{1}{3}$ means that the "line of grays" is no longer a line but a curve, which greatly complicates the math with no real practical benefit for most applications.

Using Chroma Instead of Saturation

In reality, saturation does not do a good job of capturing the purity of a color. Recall that both *HSV* and *HSL* were devised for the context of computer graphics and user interfaces, in which it makes sense for the quantity presented to the user (saturation) to always range from 0 to 1. For image processing and analysis, however, it is more important that the quantity capture some notion of the color purity. To see that saturation does not do this, consider the color $(R', G', B') = (0.01, 0, 0)$, assuming that the values range from 0 to 1. On the screen such a color would be considered black and colorless, but its saturation is 1 in both *HSV* and *HSL*, as if it were completely saturated. To overcome this problem, it is often preferable to use the chroma, which is obtained by removing the normalization, resulting in color spaces that could be called *HCV* or *HCL*. There are four choices for chroma, any of which may be used: the actual distances C and $\overline{C}$ in Equations (9.82) and (9.93), respectively, or their approximations $C' \equiv V_\delta$ in the case of *HSV* or $\overline{C}' \equiv 1 - V_{min}$ in the case of *HSL*. Figure 9.16 shows an example of chroma versus saturation for a real image.

9.5.9 CIE *XYZ*, *L* u* v**, and *L* a* b**

In many applications we would like to be able to measure the difference between two colors. Clearly, such differences only make sense when they are small, because (for example) it makes no sense to ask whether red looks more like green than blue looks like yellow. Nevertheless, even with small differences, the choice of color space is important. In particular, one cannot expect to obtain a meaningful result by performing such a difference in an *RGB* or *R'G'B'* space. To see that CIE *XYZ* has the same limitation, note that in Figure 9.10 green occupies a much larger area than yellow, so that the distance between two colors

[4] As before, the derivation is omitted.

Original HSV HSL Max − min

Figure 9.16 Saturation versus chroma. Chroma more consistently captures the intuitive notion of the pureness of the color than does saturation.

that are very far from each other (e.g., red and yellow) can be the same as the distance between two colors that look quite similar (e.g., two types of green). Quantitatively, such a result can be obtained by performing a variation of the color matching experiment in which two color patches are presented to the observer under the same lighting, and the observer is asked whether the two patches are the same color. The threshold for being able to distinguish between two colors is known as the just-noticeable difference (JND),[†] and if the contours of the JND regions are plotted on the CIE chromaticity diagram, they appear as ellipses which are known as **MacAdam ellipses**.

In a perceptually uniform color space the JND contours are circles, so that Euclidean distances in the color space capture the perceptual dissimilarity between colors. One way to achieve an approximation to such an ideal is to apply a projective transform to the XYZ coordinates:

$$u' \equiv \frac{4X}{X + 15Y + 3Z} = \frac{4x}{3 - 2x + 12y} \tag{9.97}$$

$$v' \equiv \frac{9Y}{X + 15Y + 3Z} = \frac{9y}{3 - 2x + 12y} \tag{9.98}$$

where (u', v') define the **CIE 1976 Uniform color space (UCS)**. The quantities u' and v' denote the CIE 1976 successors to the obsolete 1960 CIE u and v quantities, related by $u = u'$, $v = \frac{2}{3}v'$.

The CIE 1976 UCS is used by the **CIE 1976 $L^*u^*v^*$ color space (CIELUV)**.[‡] The transformation from XYZ to $L^*u^*v^*$ is given by applying a nonlinear transfer function to the relative luminance, then multiplying by shifted versions of u' and v',

$$L^* \equiv 116 f(Y/Y_n) - 16 \tag{9.99}$$

$$u^* \equiv 13L^*(u' - u'_n) \tag{9.100}$$

$$v^* \equiv 13L^*(v' - v'_n) \tag{9.101}$$

[†] Section 2.1.2 (p. 21).

[‡] CIELUV is written without any space and is pronounced "SEA-love". Similarly, CIELAB is pronounced "SEA-lab".

where the relative luminance is obtained by dividing the CIE luminance Y by the luminance Y_n of the reference white, and where the nonlinear transfer function f is defined as the cube root for most values, with a linear slope at low intensities chosen so that the function and its derivative are continuous at the junction:

$$f(t) = \begin{cases} t^{1/3} & \text{if } 0.008856 \le t \le 1 \\ 7.787t + 16/116 & \text{if } 0 \le t < 0.008856 \end{cases} \tag{9.102}$$

The values u'_n and v'_n are the u' and v' of the reference white, whose tristimulus values are given by X_n, Y_n, and Z_n, respectively. We should note that the shifting of u' and v' in $L^*u^*v^*$ has no basis in psychovisual experiments, so there is no reason to expect them to perform well in scenarios far from the white point used to derive them. The quantity L^*, known as CIE "lightness" and ranging from 0 to 100, is the standard approximation to the human perceptual response to luminance, designed by taking into account Stevens' Power Law.[1] It closely models the Munsell value, except that the range is different.

An alternative to CIELUV is the **CIE 1976 $L^*a^*b^*$ color space (CIELAB)**, which is also designed to be perceptually uniform. Rather than using a projective transform, as in CIELUV, CIELAB relies upon color differences similar to opponent colors:

$$L^* \equiv 116f(Y/Y_n) - 16 \tag{9.103}$$

$$a^* \equiv 500\,(f(X/X_n) - f(Y/Y_n)) \tag{9.104}$$

$$b^* \equiv 200\,(f(Y/Y_n) - f(Z/Z_n)) \tag{9.105}$$

where f is defined in Equation (9.102). The a^* axis roughly corresponds to the red-green opponent colors, while the b^* axis corresponds to the blue-yellow opponent colors. Although the committee at the time (in 1976) could not agree between the two alternatives for chroma (hence resulting in two standards), over the years it has generally been agreed upon that $L^*a^*b^*$ is the more accurate model, so that $L^*a^*b^*$ is the standard approach today when a perceptually uniform color space is needed. Even so, note by comparing with Equation (9.31) that $L^*a^*b^*$ itself is flawed because it is based upon a "wrong von Kries" method, with scaling performed (incorrectly) in XYZ space rather than in the space of the L-, M-, and S-cones.

The total color difference $\sqrt{(L_1^* - L_2^*)^2 + (u_1^* - u_2^*)^2 + (v_1^* - v_2^*)^2}$ in $L^*u^*v^*$ space is approximately perceptually uniform. That is, two colors whose difference is small are not perceptually distinct, whereas two colors whose difference is large are noticeably different. Similarly, in $L^*a^*b^*$ the total color difference is given by $\sqrt{(L_1^* - L_2^*)^2 + (a_1^* - a_2^*)^2 + (b_1^* - b_2^*)^2}$. More recent formulations, such as **CIE94** and **CIEDE2000**, include additional normalization and terms to improve perceptual normalization.

9.5.10 CMYK

The final color space we shall consider is $CMYK$ (cyan-magenta-yellow-black). Unlike all the other color spaces we have mentioned, CMYK is a subtractive color space used for printing. In theory, it is simply the additive inverse of RGB:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = 1 - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{9.106}$$

[1] Section 2.3.2 (p. 43).

so that pure red ($R = 1, G = B = 0$) is a mixture of magenta and yellow, and so forth, a relationship that is obvious from the color wheel and chromaticity diagram. In practice, however, as we have mentioned, paint pigments and inks are highly nonlinear, and therefore subtractive color spaces are much more difficult to characterize mathematically. Although we often say that cyan, magenta, and yellow are the primary colors of printing, this is really just an approximation necessitated by economics. Inexpensive printers can faithfully reproduce a wide range of colors using just mixtures of these three colors, but the color gamut of such printers (which are not triangles but rather irregular shapes) tend to be much smaller than the gamut of displays. As a result, high-end printers use more than three primary colors (e.g., orange and green), and paint mixers in paint stores typically combine 12 or more colors.

9.6 Further Reading

The material in this chapter lies at the intersection of color science (colorimetry), video engineering, and image processing. There are many good books on color science, such as the classic by Wyszecki and Stiles [1982] or the more recent book by Lee [2005]. An excellent short introduction to color appearance in general has been written by Harold [2001] as well as Brainard and Stockman [2009], and another easily accessible presentation can be found in the book by Fairchild [2011]. Such works cover the CIE chromaticity diagram, CIE color spaces, CAMs and CATs, the difference between SSFs and CMFs, and the difference between saturation, chroma, and colorfulness. Helpful descriptions can also be found in Wandell [1995] and Palmer [1999], which are written from a psychological point of view. For an introductory presentation aimed at the prepress and printing industries, see Green [1999]. And, of course, the book that started it all is the classic work of Newton [1704].

Early work on color matching is due to Grassmann [1854]. Grassmann influenced Maxwell [1860], who performed detailed color matching experiments of his own, providing an early description of metamers. The CIE 1931 CMFs are based on the 2° experiments conducted independently by Wright [1929] and Guild [1931]. However, these experiments did not measure the CMFs directly, so the CIE 1931 CMFs had to be obtained by assuming that the CIE 1924 photopic LEF was a linear combination of the CMFs, which led to errors in the estimated CMFs due to the fact that the CIE 1924 LEF curve has a noticeable error in the small-wavelength (blue) region. As explained by Stockman and Sharpe [1999], corrections later made by Judd [1951] and Vos [1978] alleviate this problem. The CIE 1964 CMFs are based on the 10° experimental results presented by Stiles and Burch [1959] and, to a lesser extent, on those by Speranskaya [1959]. The cone SSFs were first measured by Svaetichin [1956]. The CMFs shown in Figure 9.5 are from the 2° experiments of Stiles and Burch [1955]; while the cone SSFs in Figure 2.2 are from Stockman and Sharpe [2000] and are based on the 10° data of Stiles and Burch [1959], among other sources. The data for the CMFs, SSFs, and CIE chromaticity diagram can be found online.[7]

Some of the material in this chapter relies upon the video engineering insights of Poynton [2003]. In that work the reader will find descriptions on transforming between cameras and displays, the mathematics of white balancing, the distinction between luminance and luma, and so forth. The diagram in Figure 9.12 explaining the origin of luma is adapted from that work, and the notation ${}^{601}_{255}Y'$ can also be found there. The distinction between the two terms luma and luminance were clarified in Engineering Guideline EG 28, which was adopted by SMPTE in 1993. The official Rec. 601, Rec. 709, and Rec. 2020 specifications can be found online.[8] The term "wrong von Kries" is due to Terstiege [1972].

The Munsell Color System was first described in the classic books of Munsell [1905, 1915]. The so-called Hanbury transformation is from Hanbury [2008]. The Bradford transform is due to Lam [1985], while the Sharp transform can be found in Finlayson and Süsstrunk [2000]. The Macbeth ColorChecker is the work of McCamy et al. [1976]. Alvy Ray Smith is the pioneer behind *HSV* and *HSL*, and the equations presented here are from his original publication of Smith [1978]. In particular, our HSL model is identical to Smith's original HSL model (except for our restriction on the luma coefficients); it is also identical to the model called HSI in Gonzalez and Woods [2008]. For measuring the difference between color distributions, see the earth mover's distance as described by Rubner et al. [1998].

[7] Colour and Vision Research Laboratory, http://www.cvrl.org

[8] http://www.itu.int