

Reference Input/Output Medium Metric RGB Color Encodings (RIMM/ROMM RGB)

Kevin E. Spaulding, Geoffrey J. Woolfe and Edward J. Giorgianni
Eastman Kodak Company
Rochester, New York, U.S.A.

Abstract

A new color encoding specification known as Reference Output Medium Metric RGB (*ROMM RGB*) is defined. This color encoding is intended to be used for storing, interchanging and manipulating images that exist in a rendered image state without imposing the gamut limitations normally associated with device-specific color spaces. *ROMM RGB* was designed to provide a large enough color gamut to encompass most common output devices, while simultaneously satisfying a number of other important criteria. It is defined in a way that is tightly linked to the ICC profile connection space (PCS) and is suitable for use as an Adobe Photoshop™ working color space. A companion color encoding specification, known as Reference Input Medium Metric RGB (*RIMM RGB*), is also defined. This encoding can be used to represent images in an unrendered scene image state.

Introduction

Digital images are often encoded in terms of color spaces that are tied directly to the characteristics of actual input or output devices. Common examples of such color spaces are scanner RGB, video RGB, and CMYK. However, such spaces generally are *device-dependent* in that their values can be associated with specific colorimetric values only in the context of the characteristics of the particular device on which the image is displayed or captured.

On the other hand, *device-independent* color spaces generally are meant to represent colorimetric values directly. These color spaces most often are based on the system of colorimetry developed by the Commission International de l'Eclairage (CIE). Examples of such color spaces include CIE XYZ and CIELAB. It should be noted that the specification of a color value in a device-independent (or device-dependent) color space does not fully specify color *appearance* unless the viewing conditions also are known. For example, two patches with identical colorimetric values can have very different color appearance, depending on the conditions under which they are viewed.

The fact that images exist in many different color spaces significantly complicates the development of software applications that use and manipulate images. For example, an image-processing algorithm that works in one color space might not have the expected behavior when used in another color space. This has led many people to advocate the use of a standard color encoding (or perhaps a small number of standard color encodings) for the storage, interchange and manipulation of digital images. Often, these proposals have involved specifying a particular output-device-dependent color space to be a "standard." Examples of such color spaces include *SWOP CMYK*[1] and *sRGB*.[2]

One significant problem with specifying an output-device-dependent color space as the standard is that typically it will limit the encodable color gamut and luminance dynamic range of images according to the capabilities of a specific output device. For example, hardcopy media and CRT displays typically have very different color gamuts. Therefore, using *sRGB* (which is based on a particular CRT model) as a standard color encoding would necessarily involve clipping many colors that could have been produced on a given hardcopy medium.

The International Color Consortium (ICC)[3] has defined a Profile Connection Space (PCS) that comprises a device-independent color encoding specification that can be used to explicitly specify the color of an image with respect to a reference viewing environment. *Device profiles* can be used in a color management system to relate the device-dependent code values of input images to the corresponding color values in the PCS, and from there to the device-dependent output color values appropriate for a specific output device. It could be argued that the PCS could serve as the standard color encoding we are looking for. However, it was never intended that the PCS be used to directly store or manipulate images. Rather, it was simply intended to be a color space where profiles could be joined to form complete input-to-output color transforms. Neither the CIELAB nor the XYZ color encodings supported for the PCS is particularly well suited for many common kinds of image manipulations. It is therefore desirable to define a standard large-gamut color encoding that can be used for storing, interchanging and manipulating color images. This paper will describe a new color space known as *Reference Output Medium Metric RGB* (*ROMM RGB*). This color encoding is tightly coupled to the ICC PCS and is intended to be used for encoding *rendered output images* in a device-independent fashion.

Rendered output images should be distinguished from images that are intended to be an encoding of the colors of an *original scene*. It is well known that the colorimetry of a pleasing rendered image does not match the colorimetry of the corresponding scene. Among other things, the tone/color reproduction process that "renders" the colors of a scene to the desired colors of the rendered image must compensate for differences between the scene and rendered image viewing conditions.[4,5] For example, rendered images generally are viewed at luminance levels much lower than those of typical outdoor scenes. As a consequence, an increase in the overall contrast of the rendered image usually is required in order to compensate for perceived losses in reproduced luminance and chrominance. Additional contrast increases in the shadow regions of the image also are needed to compensate for viewing flare associated with rendered-image viewing conditions.

In addition, psychological factors such as color memory and color preference must be considered in image rendering. For example, observers generally remember colors as being of higher purity, and they typically prefer skies and grass to be more colorful than they were in the original scene. The tone/color reproduction aims of well-designed imaging systems are designed to account for such factors.

Finally, the tone/color reproduction process also must account for the fact that the dynamic range of a rendered image usually is substantially less than that of an original scene. It is therefore necessary to discard and/or compress some of the highlight and shadow information of the scene to fit within the dynamic range of the rendered image.

Due to these and other factors, color encodings such as *ROMM RGB* that are intended for encoding rendered output images are inappropriate for use in encoding original-scene images. Rather, a color encoding that is directly related to the color of an original scene should be used. Accordingly, a companion to the *ROMM RGB* color encoding specification, known as *Reference Input Medium Metric (RIMM RGB)*, has also been defined. This encoding is intended to represent original scene color appearance. The *RIMM RGB* color encoding not only provides extra dynamic range necessary for the encoding of scene information, it also provides a mechanism for clearly distinguishing whether or not an image has been rendered.

Selection of Color Space

It is desirable that the *RIMM RGB* and *ROMM RGB* color encoding specifications be defined such that they are as similar as possible to one another. This simplifies the development of image-manipulation algorithms across the two color encodings. It also simplifies the rendering process in which a rendered *ROMM RGB* image is created from an original scene image encoded in *RIMM RGB*. This is best achieved by basing the two encodings on the same color space. The criteria that were used to select this color space include the following:

- Direct relationship to the color appearance of the scene/image
- Color gamut large enough to encompass most real-world surface colors
- Efficient encoding of the color information to minimize quantization artifacts
- Simple transformation to/from ICC PCS
- Simple transformation to/from video RGB (e.g., *sRGB*)
- Well-suited for application of common image manipulations such as tonescale modifications, color-balance adjustments, sharpening, etc.
- Compatible with established imaging workflows

An additive RGB color space with an appropriately selected set of "big RGB" primaries is ideal for satisfying all of these criteria. When images are encoded using any such set of primaries, there is a direct and simple relationship to scene/image colorimetry because the primaries are linear transformations of the CIE XYZ primaries. Big RGB color spaces have the additional advantage that simple LUT-matrix-LUT transformation can be used to convert to/from additive color spaces such as PCS XYZ, video RGB (*sRGB*) and digital camera RGB.

Two of the criteria applied that affect the selection of RGB primaries are somewhat conflicting. First, their chromaticities should define a color gamut sufficiently large to encompass colors likely to be found in real scenes/images. At the same time, their use should result in efficient digital encodings that minimize quantization errors.

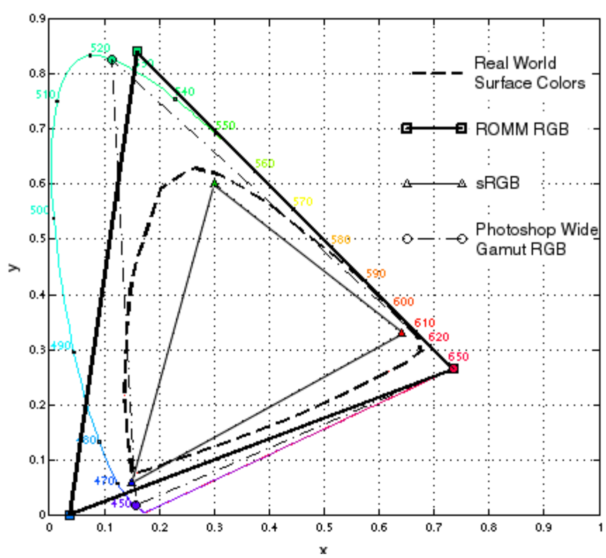

Figure 1. Comparison of primaries in x-y chromaticity coordinates

Increasing the gamut can only be achieved by trading off against correspondingly larger quantization errors. If the primaries are chosen to include the maximum possible chromaticity gamut (i.e., the entire area within the spectrum locus), a significant fraction of the color space would correspond to imaginary colors located outside that region. Therefore, in any encoding using such a color space, there would be “wasted” code values that would never be used in practice. This would lead to larger quantization errors in the usable part of the color space than would be obtained with different primaries defining a smaller chromaticity gamut. It is therefore desirable to choose primaries with a gamut that is “big enough” but not “too big.”

Figure 1 shows the primaries selected for *RIMM/ROMM RGB*. Clearly, these primaries encompass the gamut of real world surface colors, without devoting a lot of space to non-realizable colors outside the spectrum locus. Also shown for comparison are the *sRGB* primaries. It can be seen that the *sRGB* color gamut is inadequate to cover significant portions of the real world surface color gamut. In particular, it misses many important saturated colors near the yellow-to-red boundary of the spectrum locus. The default Photoshop Wide Gamut RGB gamut, which is also shown on the figure, misses some of these colors as well.

One of the important requirements for *RIMM/ROMM RGB* is that they be well suited for application of common image manipulations. Many types of common image manipulations include the step of applying non-linear transformations to each of the channels of an RGB image (e.g., tonescale modifications, color balance adjustments, etc.). The process of forming a rendered image from a scene is one important application of this type. One way to accomplish the rendering operation is by means of applying various nonlinear transforms to the individual channels of an RGB image. These transforms can result in several desirable color/tone reproduction modifications, including:

- Increasing luminance and color contrast in mid-tones and compressing contrast of highlights and shadows.
- Increasing the chroma of in-gamut colors.
- Gamut mapping out-of-gamut colors in a simple but visually pleasing way.

If an input scene is represented using the *RIMM RGB* color encoding, the result of applying such rendering transforms will be a rendered image in the *ROMM RGB* color encoding.

The nonlinear transforms used in rendering will, in general, modify the relative ratios of the red, green and blue channel data. This can lead to hue shifts, particularly for highly saturated colors. Hue shifts are particularly problematic when they occur in a natural saturation gradient within an image. Such gradients tend to occur when rounded surfaces are illuminated by a moderately directional light source. In such situations, chroma increases with distance from the specular highlight and then decreases again as the shadows deepen.

There is a tradeoff between the color gamut of the primaries, quantization artifacts, and the extent of the hue shifts that occur during rendering. If the primaries are moved out so as to increase the color gamut, quantization artifacts will increase and the hue shifts introduced during the application of a nonlinear transformation will decrease. This results from the fact that the RGB values will be clustered over a smaller range, thereby reducing the impact of nonlinear transformations. If the color gamut is decreased by moving the primaries closer together, quantization artifacts diminish but hue shifts are generally larger and color gamut is sacrificed. During the selection of the *RIMM/ROMM RGB* primaries, an extensive optimization process was used to determine the best overall solution.

Such hue shifts can never be completely eliminated, so the objective when optimizing the location of the primaries was to eliminate or minimize objectionable hue shifts at the expense of less noticeable or less likely hue shifts. Hue shifts for a particular color can be eliminated when the color lies on one of the straight lines passing through the primaries and the white point on a chromaticity diagram.

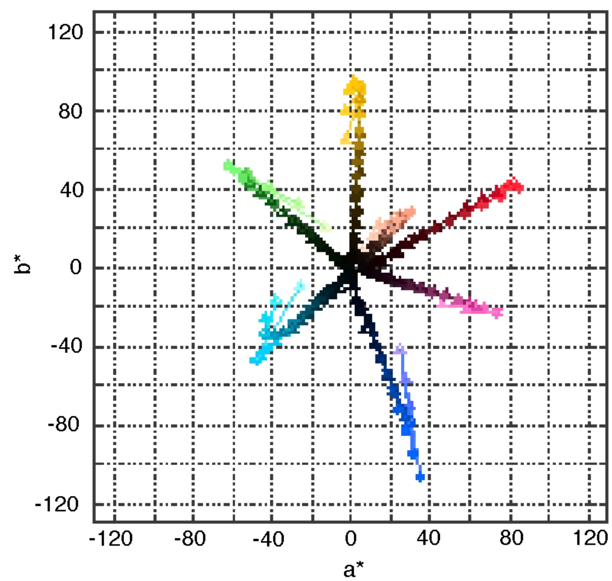

Figure 2. Hue shifts for the RIMM/ROMM RGB color encoding resulting from a typical nonlinear rendering transform.

Hue shifts introduced by the application of nonlinear transformations were examined during the process of selecting the *RIMM/ROMM RGB* primaries by studying a chroma series for eight color patches from the Macbeth Color Checker™. These patches included red, yellow, green, cyan, blue, magenta, light flesh and dark flesh. Hue shifts in flesh tones and yellows, particularly in the direction of green, are considered to be the most objectionable. These hue shifts are most strongly affected by the location of the blue primary. As a consequence, the location of the blue primary is constrained by the need to minimize undesirable hue shifts and maximize the color gamut of the *RIMM/ROMM RGB* color encoding. Other colors that were considered to be particularly important during the optimization process were blues and reds. The hue shifts associated with the selected *RIMM/ROMM RGB* primaries are shown in Fig. 2. This plot shows a series of line segments connecting the a*, b* values before and after a nonlinear tonescale was applied to a chroma series in each of the eight color directions. It can be seen that small hue shifts are introduced for the most saturated colors in the blue and cyan directions, but that the hue shifts elsewhere are quite small. The resulting hue shifts associated with the primaries of the default Adobe Photoshop Wide Gamut RGB color space are shown in Fig. 3. It can be seen that the hue shifts for these primaries are significantly larger than those of the *RIMM/ ROMM RGB* primaries in almost every color region.

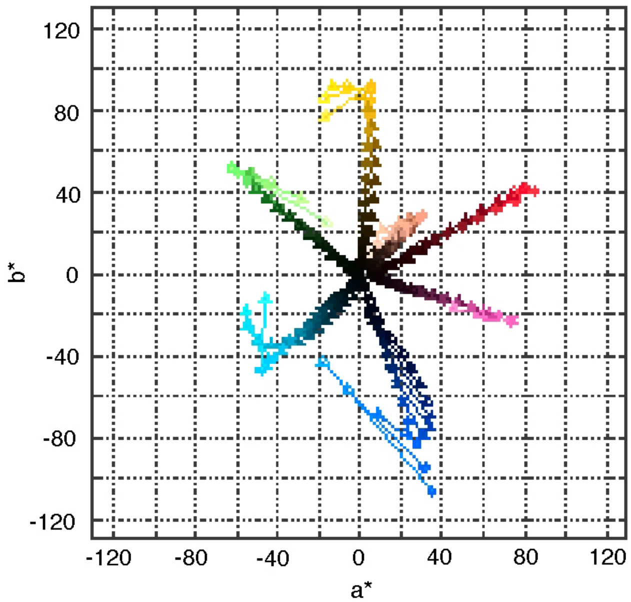

Figure 3. Hue shifts for default Adobe Photoshop Wide Gamut RGB color space resulting from a typical nonlinear rendering transform.

Finally, a basic requirement for any commercially useful color encoding is that it be compatible with typical commercial imaging workflows. In many cases, Adobe Photoshop software is an important component in such imaging chains. Conveniently, the latest version of Adobe Photoshop has incorporated the concept of a "working color space," which is different from the monitor preview color space. This is very consistent with the notion of storing/manipulating images in a "big RGB" color space. Adobe has placed a constraint on the definition of valid working color spaces that requires the primaries to have all positive *x-y-z* chromaticity values. This implies that the primaries must be inside the triangle defined by the points (0,0), (1,0) and (0,1) on Fig. 1. This condition is satisfied for the *ROMM RGB* primaries. (Since Photoshop operates within a rendered-image paradigm, it is inappropriate to use *RIMM RGB* as a Photoshop working color space.)

Definition of *ROMM RGB*

In addition to defining a color space, it is also necessary to specify an intended viewing environment in order to unambiguously define a color-appearance encoding. One of the requirements for *ROMM RGB* is that it be tightly coupled to the ICC Profile Connection Space (PCS). Color values in the PCS represent the CIE colorimetry of an idealized reference medium that will produce the desired color appearance when viewed in a reference viewing environment. Eastman Kodak Company has proposed a specific encoding reference viewing environment that can be used to unambiguously define the PCS for the purposes of producing ICC profiles.[6,7] This reproduction viewing environment is defined to have the following characteristics:

- Luminance level is in the range of 160-640 cd/m^2.
- Viewing surround is average.
- There is 0.5-1.0% viewing flare.
- The adaptive white point is specified by the chromaticity values for CIE Standard Illuminant D50 (x = 0.3457, y = 0.3585).
- The image color values are assumed to be encoded using flareless (or flare corrected) colorimetric measurements based on the CIE 1931 Standard Colorimetric Observer.

The *Reference Output Medium Metric RGB* (*ROMM RGB*) color encoding is defined in the context of this viewing environment by the color values associated with a hypothetical additive color device having the following characteristics:

- Reference primaries defined by the CIE chromaticities given in Table 1.
- Equal amounts of the reference primaries produce a neutral with the chromaticity of D50.
- The capability of producing a black with L* = 0.
- No cross-talk among the color channels (i.e., red output is affected only by red input, green output is affected only by green input, and blue output is affected only by blue input).

Table 1. Primaries/white point for Reference Output Medium.

Color	x	y
Red	0.7347	0.2653
Green	0.1596	0.8404
Blue	0.0366	0.0001
White	0.3457	0.3585

Additionally, a quantization scheme must be specified to store the *ROMM RGB* values in an integer form. A simple gamma function nonlinearity incorporating a slope limit is defined for this purpose supporting 8-bit/channel, 12-bit/channel, and 16-bit/channel quantization schemes.

The conversion of the PCS XYZ tristimulus values to *ROMM RGB* values can be performed by a matrix operation, followed by a set of 1-D functions. This is equivalent to the operations associated with a basic CRT profile. This means that *ROMM RGB* can be used conveniently in a system employing ICC profiles using an appropriately designed monitor profile.

ROMM RGB Conversion Matrix

Given the defined primaries shown in Table 1, the following matrix can be derived to compute the linear *ROMM RGB* values from the PCS rendered image tristimulus values:

$$\begin{bmatrix} R_{ROMM} \\ G_{ROMM} \\ B_{ROMM} \end{bmatrix} = \begin{bmatrix} 1.3460 & -0.2556 & -0.0511 \\ -0.5446 & 1.5082 & 0.0205 \\ 0.0000 & 0.0000 & 1.2123 \end{bmatrix} \begin{bmatrix} X_{PCS} \\ Y_{PCS} \\ Z_{PCS} \end{bmatrix}, \quad (1)$$

where it is assumed that the PCS tristimulus values have been scaled so that the Y_{PCS} value for the idealized reference medium is 1.0. As required by the definition of the *ROMM RGB*, this matrix will map image tristimulus values with the chromaticity of D50 to equal *ROMM RGB* values. A neutral with a Y_{PCS} value of 1.0 will map to linear *ROMM RGB* values of 1.0. These unity *ROMM RGB* values will therefore correspond to the white point of the idealized reference medium associated with the ICC PCS.

Nonlinear Encoding of *ROMM RGB*

The functional form of the *ROMM RGB* nonlinearity is a gamma function with a linear segment at the dark end of the intensity scale:

$$X'_{ROMM} = \begin{cases} 0; & X_{ROMM} < 0.0 \\ 16\ X_{ROMM}\ I_{\max}; & 0.0 \le X_{ROMM} < E_t \\ \left(X_{ROMM}\right)^{1/1.8} I_{\max}; & E_t \le X_{ROMM} < 1.0 \\ I_{\max}; & X_{ROMM} \ge 1.0 \end{cases}, \quad (2)$$

where X is either R, G, or B, $I_{\max}$ is the maximum integer value used for the nonlinear encoding, and

$$E_t = 16^{1.8/(1-1.8)} = 0.001953 \ . \quad (3)$$

For the baseline 8-bit configuration, $I_{\max}$ is equal to 255. The linear segment of the nonlinearity is used to impose a slope limit so as to minimize reversibility problems because of the infinite slope of the gamma function at the zero point. 12- and 16-bit versions of *ROMM RGB* are also defined. The only difference is that the value of $I_{\max}$ is set to 4095 or 65535, respectively. In cases where it is necessary to identify a specific precision level, the notation *ROMM8 RGB*, *ROMM12 RGB* and *ROMM16 RGB* is used. Table 2 shows some sample encodings for a series of neutral patches of specified relative image intensity, where a relative image intensity of 1.0 corresponds to the white point of the PCS.

Table 2. Sample neutral patch encodings.

Relative Intensity	PCS L*	ROMM8 RGB	ROMM12 RGB	ROMM16 RGB
0.00	0.00	0	0	0
0.001	0.90	4	66	1049
0.01	8.99	20	317	5074
0.10	37.84	71	1139	18236
0.18	49.50	98	1579	25278
0.35	65.75	142	2285	36574
0.50	76.07	174	2786	44590
0.75	89.39	217	2490	55855
1.00	100.00	255	4095	65535

One application of the *ROMM RGB* color encoding is as a working color space for Adobe Photoshop software. It should be noted that Adobe Photoshop software currently limits the nonlinearity that can be used to define a valid working space to be a simple gamma function. However, both the Adobe Photoshop software implementation and the *Kodak Digital Science*™ Color Matching Module (CMM) implementation automatically impose a *slope limit* of 16 at the dark end of the tone scale. Although a profile that explicitly incorporates the nonlinearity with the slope limit can not be used by Adobe Photoshop software, a profile using a simple gamma function nonlinearity produces the net effect of Eq. (2) when used by Adobe Photoshop software or the current version of the Kodak CMM. Therefore, to ensure Adobe Photoshop software compatibility, the *ROMM RGB* ICC profile created by Eastman Kodak Company uses a simple gamma function nonlinearity without the slope limit, rather than the form shown in Eq. (2). At some point in the future it may be possible to produce a new ICC profile that explicitly incorporates the slope limit if Adobe were to modify the Photoshop software to remove this restriction. Although this would not have any effect on the results obtained using Adobe Photoshop software or the Kodak CMM, it would increase the likelihood that equivalent results would be obtained using different CMMs that may or may not include the same slope limiting feature.†

Inverse of *ROMM RGB* Encoding

It is also necessary to define an inverse transformation to convert *ROMM RGB* values back to rendered image PCS values. This can be done by simply inverting the nonlinear function given in Eq. (2), and then applying the inverse of the matrix given in Eq. (1).

The first step is to undo the nonlinear encoding of the *ROMM RGB* values. This will convert the signals back to linear *ROMM RGB* values.

† For more information about using *ROMM RGB* as a Photoshop working space, see the white paper posted at www.Kodak.com (search on "ROMM").

$$X_{ROMM} = \begin{cases} \dfrac{X'_{ROMM}}{16\ I_{\max}}; & 0.0 \le X'_{ROMM} < 16\ E_t\ I_{\max} \\ \left(\dfrac{X'_{ROMM}}{I_{\max}}\right)^{1.8}; & 16\ E_t\ I_{\max} \le X'_{ROMM} \le I_{\max} \end{cases}, \quad (4)$$

where X_{ROMM} and X'_{ROMM} are the nonlinear and linear *ROMM RGB* values, respectively, and as before, X is either R, G or B.

To convert the *ROMM RGB* values to the corresponding D50 PCS tristimulus values, it is simply necessary to multiply by the inverse of the matrix given in Eq. (1)

$$\begin{bmatrix} X_{PCS} \\ Y_{PCS} \\ Z_{PCS} \end{bmatrix} = \begin{bmatrix} 0.7977 & 0.1352 & 0.0313 \\ 0.2880 & 0.7119 & 0.0001 \\ 0.0000 & 0.0000 & 0.8249 \end{bmatrix} \begin{bmatrix} R_{ROMM} \\ G_{ROMM} \\ B_{ROMM} \end{bmatrix}. \quad (5)$$

As expected, when this matrix is applied to linear *ROMM RGB* values that are equal, tristimulus values with the chromaticity of D50 are obtained.

Conversion Between *ROMM RGB* and *sRGB*

In many cases, it will be necessary to convert *ROMM RGB* values to a video RGB representation for display on a CRT. This can be accomplished by combining the *ROMM RGB* to PCS transformation described in the previous section with an appropriate PCS to video RGB transformation for the CRT. Consider the special case of a CRT that responds according to the *sRGB* specification.[2] Because *sRGB* is defined using a D65 white point, and the PCS is defined using a D50 white point, the first step in the conversion of PCS values to *sRGB* values must be a D50-to-D65 chromatic adaptation. This can be accomplished using a simple von Kries transformation as follows.

$$\begin{bmatrix} X_{D65} \\ Y_{D65} \\ Z_{D65} \end{bmatrix} = \begin{bmatrix} 0.9845 & -0.0547 & 0.0678 \\ -0.0060 & 1.0048 & 0.0012 \\ 0.0000 & 0.0000 & 1.3200 \end{bmatrix} \begin{bmatrix} X_{D50} \\ Y_{D50} \\ Z_{D50} \end{bmatrix}. \quad (6)$$

Alternatively, other chromatic adaptation transforms could also be used.

The *sRGB* color space is defined using the phosphor primaries associated with Rec. 709. The conversion from D65 tristimulus values to the linear RGB values associated with these primaries is given by the following inverse phosphor matrix:

$$\begin{bmatrix} R_S \\ G_S \\ B_S \end{bmatrix} = \begin{bmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X_{D65} \\ Y_{D65} \\ Z_{D65} \end{bmatrix}. \quad (7)$$

Finally, the desired *sRGB* code values can be computed by applying the appropriate nonlinearity and integerizing:

$$X'_S = \begin{cases} 255(12.92 X_S); & X_S \le 0.0031308 \\ 255(1.055 X_S^{1/2.4} - 0.055); & X_S > 0.0031308 \end{cases}, \quad (8)$$

where X is either R, G, or B.

Conversion from *ROMM RGB* values to the *sRGB* code values can therefore be accomplished by applying the inverse *ROMM RGB* nonlinearity given in Eq. (4), followed by the matrices given in Eqs. (5), (6) and (7), followed by the *sRGB* nonlinearity given in Eq. (8). The three sequential matrix operations can be combined by cascading the matrices together to form the following single matrix:

$$\begin{bmatrix} R_S \\ G_S \\ B_S \end{bmatrix} = \begin{bmatrix} 2.0564 & -0.7932 & -0.2632 \\ -0.2118 & 1.2490 & -0.0372 \\ -0.0152 & -0.1405 & 1.1556 \end{bmatrix} \begin{bmatrix} R_{ROMM} \\ G_{ROMM} \\ B_{ROMM} \end{bmatrix}. \quad (9)$$

Thus, the transformation from *ROMM RGB* to *sRGB* can be implemented with a simple LUT-MAT-LUT chain.

It should be noted that not all colors that can be encoded in *ROMM RGB* will be within the *sRGB* color gamut. As a result, it will be necessary to perform some sort of gamut mapping to limit all of the colors to the appropriate gamut. The simplest form of gamut mapping is just to clip all of the linear *sRGB* values to the range 0.0 to 1.0 before applying the nonlinearity of Eq. (8). However, this approach can result in noticeable hue shifts in certain cases. Superior results can be obtained using more sophisticated gamut-mapping strategies.

The conversion from *sRGB* back to *ROMM RGB* is simply an inverse of the steps that were just discussed. First, the inverse of the *sRGB* nonlinearity given in Eq. (8) is applied to determine the linear RGB_s values:

$$X_S = \begin{cases} \dfrac{\left(X'_S/255\right)}{12.92}; & X'_S \le 0.04045 \times 255 \\ \left(\dfrac{\left(X'_S/255\right) + 0.055}{1.055}\right)^{2.4}; & X'_S > 0.04045 \times 255 \end{cases}. \quad (10)$$

Next, the inverse of the matrix in Eq. (9) is used to compute the linear RGB_{ROMM} values,

$$\begin{bmatrix} R_{ROMM} \\ G_{ROMM} \\ B_{ROMM} \end{bmatrix} = \begin{bmatrix} 0.5230 & 0.3468 & 0.1303 \\ 0.0892 & 0.8627 & 0.0481 \\ 0.0177 & 0.1095 & 0.8729 \end{bmatrix} \begin{bmatrix} R_S \\ G_S \\ B_S \end{bmatrix}. \quad (11)$$

Finally, the *ROMM RGB* nonlinearity given in Eq. (2) is applied to determine the *ROMM RGB* values.

As noted above, many colors that can be represented in *ROMM RGB* color encoding are outside the gamut of *sRGB*. As a result, the process of mapping an image from *ROMM RGB* to *sRGB* and back again generally is not lossless. Therefore, it should be emphasized that, whenever possible, a video RGB color space should not be used as an intermediate color space during the process of manipulating a *ROMM RGB* image. Rather, the image manipulations should be applied to the *ROMM RGB* image directly, and the *ROMM RGB* to *sRGB* transformation should be used to provide an image for video preview purposes only.

On the other hand, if an original image is in a video RGB color space, it should be possible to convert the image to *ROMM RGB* for manipulation purposes, and then convert it back to the video RGB color space again with only minimal losses due to quantization effects. These effects can be reduced to negligible levels by using the 12-bit/channel or 16-bit/channel versions of *ROMM RGB*. However, it should be noted that if the manipulation process creates any color values that are outside the video RGB gamut, these values will be clipped when the processed image is converted back to the original color space.

Definition of *RIMM RGB*

RIMM RGB is a companion color encoding specification to *ROMM RGB* that can be used to encode the colorimetry of an *unrendered scene*. Both encodings utilize the same big RGB color space with the primaries and white point given in Table 1. The reference viewing conditions used to encode scene color values for *RIMM RGB* are typical of outdoor environments, and are defined as follows:

- Scene luminance level is >1,600 cd/m^2.
- Viewing surround is average. (In other words, the overall luminance level and chrominance of the surround is assumed to be similar to that of the scene.)
- There is no viewing flare for the scene.
- The observer adaptive white-point is specified by the chromaticity values for D50: x = 0.3457 and y = 0.3585. (Scenes captured under illuminants different from D50 should be corrected to D50 using a chromatic adaptation transform before encoding them in *RIMM RGB*)
- The scene color values are assumed to be encoded using flareless (or flare corrected) colorimetric measurements based on the CIE 1931 Standard Colorimetric Observer.

RIMM RGB Conversion Matrix

Since *ROMM RGB* and *RIMM RGB* use a common color space, the conversion from the scene tristimulus values to the corresponding linear *RIMM RGB* values can be accomplished using the same conversion matrix that was given in Eq. (1).

Nonlinear Encoding of *RIMM RGB*

Since the dynamic range of unrendered scenes is generally larger than that of the idealized reflection print medium specified for *ROMM RGB*, a different nonlinear encoding must be used. The *RIMM RGB* nonlinearity is based on that specified by Recommendation ITU-R BT.709 (Rec. 709).[8] (This recommendation was formerly known as CCIR 709.) This is the same nonlinearity used in the *PhotoYCC* Color Space encoding implemented in the *Kodak Photo CD* System,[5] and is given by:

$$X'_{RIMM} = \begin{cases} 0; & X_{RIMM} < 0.0 \\ \left(\dfrac{I_{\max}}{V_{clip}}\right) 4.5\ X_{RIMM}; & 0.0 \le X_{RIMM} < 0.018 \\ \left(\dfrac{I_{\max}}{V_{clip}}\right)\left(1.099 X_{RIMM}^{0.45} - 0.099\right); & 0.018 \le X_{RIMM} < E_{clip} \\ I_{\max} & X_{RIMM} \ge E_{clip} \end{cases} ,(12)$$

where X is either R, G, or B; $I_{\max}$ is the maximum integer value used for the nonlinear encoding; E_{clip} is the exposure level that is mapped to $I_{\max}$; and

$$V_{clip} = 1.099 E_{clip}^{0.45} - 0.099 \ . \qquad (13)$$

For the baseline 8-bit/channel *RIMM RGB* configuration, $I_{\max}$ is 255 and E_{clip} is 2.00, which corresponds to the maximum exposure level associated with 8-bit/channel encoding in the *PhotoYCC* Color Space. In this case, the corresponding value of V_{clip} would be 1.402. In some applications, it may be desirable to use a higher bit precision version of *RIMM RGB* to minimize any quantization errors. 12- and 16-bit/channel versions of *RIMM RGB* are also defined. The only difference is that the value of I_{max} is set to 4095 or 65535, respectively. In cases in which it is necessary to identify a specific precision level, the notation *RIMM8 RGB*, *RIMM12 RGB* and *RIMM16 RGB* is used.

Inverse Encoding for *RIMM RGB*

To convert from *RIMM RGB* back to the corresponding scene colorimetry, it is only necessary to invert the nonlinear encoding

$$X_{RIMM} = \begin{cases} \dfrac{V_{clip} X'_{RIMM}}{4.5\, I_{\max}}; & 0 \le X'_{RIMM} < \dfrac{0.081\, I_{\max}}{V_{clip}} \\ \left(\dfrac{\dfrac{V_{clip} X'_{RIMM}}{I_{\max}} + 0.099}{1.099}\right)^{1/0.45}; & \dfrac{0.081\, I_{\max}}{V_{clip}} \le X'_{RIMM} < I_{\max} \end{cases} ,(14)$$

and then apply the inverse matrix given in Eq. (5).

ERIMM RGB Color Encoding

The *RIMM RGB* color space is defined to have a luminance dynamic range that can encode information up to 200% of the exposure value associated with a normally exposed perfect (100%) diffuse white reflector in the scene. This should be adequate for many applications such as digital cameras. However, for some applications, most notably scanned photographic negatives, this luminance dynamic range is insufficient to encode the full range of captured scene information. For these cases, a variation of the *RIMM RGB* color space is defined, referred to as *Extended Reference Input Medium Metric RGB (ERIMM RGB)*.

As with *RIMM RGB*, *ERIMM RGB* is directly related to the colorimetry of an original scene. The nonlinear encoding function is the only encoding step that is altered. For *ERIMM RGB*, it is desirable to increase both the maximum scene exposure value that can be represented, as well as to reduce the quantization interval size. The size of the quantization interval is directly related to the minimum scene exposure value that can be accurately represented. In order to satisfy both the extended luminance dynamic range and the reduced quantization interval requirements simultaneously, it is necessary to use a higher minimum bit precision for *ERIMM RGB*. A minimum of 12-bits/color channel is recommended in this case.

Nonlinear Encoding for *ERIMM RGB*

A modified logarithmic encoding is used for *ERIMM RGB*. A linear segment is included for the very lowest exposure values to overcome the non-invertibility of the logarithmic encoding at the dark end of the tonescale. The encoding was defined such that the linear and logarithmic segments match in both value and derivative at the boundary. In equation form, this encoding is represented by

$$X'_{ERIMM} = \begin{cases} 0; & X_{ERIMM} \le 0 \\ I_{\max} \dfrac{(\log E_t - \log E_{\min})}{(\log E_{clip} - \log E_{\min})} \left(\dfrac{X_{ERIMM}}{E_t} \right); & 0 < X_{ERIMM} \le E_t \\ I_{\max} \dfrac{(\log X_{ERIMM} - \log E_{\min})}{(\log E_{clip} - \log E_{\min})}; & E_t < X_{ERIMM} \le E_{clip} \\ I_{\max}; & X_{ERIMM} > E_{clip} \end{cases} ,(15)$$

where X is either R, G, or B; I_{max} is the maximum integer value used for the nonlinear encoding; E_{clip} is the upper exposure limit that gets mapped to I_{max}; E_{min} is the lower exposure limit where the logarithmic encoding would be clipped in the absence of the linear segment; and

$$E_t = e\,E_{\min} \qquad (16)$$

is the breakpoint between the linear and logarithmic segments, e being the base of the natural logarithm. For a 12-bit encoding, I_{max} is 4095, and for a 16-bit encoding I_{max} is 65535. The appropriate values of the remaining parameters are summarized in Table. 3. In cases in which it is necessary to identify a specific precision level, the notation *ERIMM12 RGB* and *ERIMM16 RGB* is used.

Table 3. Parameter values for *ERIMM RGB*.

Parameter	value
$\log E_{clip}$	2.500
E_{clip}	316.2
Log E_{min}	-3.000
E_{min}	0.001
$\log E_t$	-2.566
E_t	0.002718

Table 4. Sample scene exposure encodings.

Relative Exposure	Rel. Log Exposure	*RIMM8 RGB*	*RIMM12 RGB*	*ERIMM12 RGB*
0.001	-3.00	1	13	119
0.01	-2.00	8	131	745
0.10	-1.00	53	849	1489
0.18	-0.75	74	1194	1679
1.00	0.00	182	2920	2234
2.00	0.30	255	4095	2458
8.00	0.90	NA	NA	2906
32.00	1.50	NA	NA	3354
316.23	2.50	NA	NA	4095

To compute ERIMM RGB values, Eq (15) should be used in place of Eq. (12) in the procedure described above for determining RIMM RGB values. Examples of RIMM RGB and ERIMM RGB encodings for neutral patches at different scene exposure levels are shown in Table 4. It can be seen that the range of exposures that can be represented in ERIMM RGB is extended relative to RIMM RGB.

Inverse Encoding for *ERIMM RGB*

The nonlinear function given in Eq. (6) can be inverted to determine an inverse *ERIMM RGB* encoding function:

$$X_{ERIMM} = \begin{cases} \dfrac{(\log E_{clip} - \log E_{\min})}{(\log E_t - \log E_{\min})} \left(\dfrac{X'_{ERIMM}\,E_t}{I_{\max}} \right); \\ \qquad \text{for } \; 0 < X'_{ERIMM} \le I_{\max} \dfrac{(\log E_t - \log E_{\min})}{(\log E_{clip} - \log E_{\min})} \\ anti\log\left[\left(\dfrac{X'_{ERIMM}}{I_{\max}} \right) (\log E_{clip} - \log E_{\min}) + \log E_{\min} \right]; \\ \qquad \text{for } \; I_{\max} \dfrac{(\log E_t - \log E_{\min})}{(\log E_{clip} - \log E_{\min})} < X'_{ERIMM} \le I_{\max} \end{cases} \quad (17)$$

where X'_{ERIMM} and X_{ERIMM} are the nonlinear and linear *ERIMM RGB* values, respectively, and as before X is either R, G, or B.

Conclusions

A family of large-gamut color encoding specifications, based on a "big RGB" color space having optimized color primaries, has been defined. *Reference Output Medium Metric RGB (ROMM RGB)* is a large-gamut device-independent color encoding designed to be used for the storage, interchange and manipulation of rendered images. It is tightly coupled to the ICC PCS, and it is compatible for use as a Photoshop working color space. *ROMM RGB* is associated with a specified encoding reference viewing environment, thereby enabling unambiguous communication of image color appearance. *Reference Input Medium Metric RGB (RIMM RGB)* is based on the same color space as *ROMM RGB* and is designed for encoding the color appearance of unrendered scenes. It is associated with a set of encoding reference viewing conditions typical of outdoor scenes. An extended dynamic range version of *RIMM RGB,* known as *ERIMM RGB*, also has been defined.. This color encoding is particularly well suited for encoding images from high-dynamic-range image sources such as color negative film. The extensive dynamic range of *ERIMM RGB* requires a minimum data precision of 12-bits/channel. Each of these color encoding specifications is based on the same big RGB color space. This facilitates the development of common image-processing algorithms and simplifies the transformations between the different color encodings.

References

1. "Graphic Technology - Color Characterization Data for Type 1 Printing," ANSI/CGATS TR 001 (1995).
2. "Multimedia Systems and Equipment – Colour Measurement and Management – Part 2-1: Colour Management – Default RGB Colour Space – sRGB," IEC 61966-2-1 (1999).
3. International Color Consortium, "File Format for Color Profiles," Specification ICC.1A:1999-04.
4. R. W. G. Hunt, The Reproduction of Colour, Fifth Edition, Fountain Press, England, 1995.
5. E. J. Giorgianni and T. E. Madden, Digital Color Management: Encoding Solutions, Addison-Wesley, Reading, MA, 1998.
6. S. Gregory, R. Poe and D. Walker, "Communicating color appearance with the ICC profile format," in IS&T and SID's 2nd Color Imaging Conference: Color Science, Systems and Applications, 170-174 (1994).
7. "Interpretation of the PCS," appendix to "Kodak ICC Profile for CMYK (SWOP) Input," ANSI CGATS/SC6 N 254, June 3, 1998
8. "Basic parameter values for the HDTV standard for the studio and for international programme exchange," Recommendation ITU-R BT.709 (formerly CCIR Recommendation 709).

Biography

Geoff Woolfe received both his BS degree (with honors) and PhD in physical chemistry from the University of Melbourne. He then worked as a post-doctoral research fellow in the field of time-resolved spectroscopy at the Technische Universitaet Muenchen (Munich, Germany) and later at the University of Melbourne, for three years prior to joining the research laboratories at Kodak Australia in 1984. In 1991 he transferred to Eastman Kodak Company in Rochester. Since his move to Rochester, he completed an MS degree in imaging science at the Rochester Institute of Technology, graduating in 1997. He is currently a Research Associate in the Imaging Science Division of Kodak's Research Laboratories. His research interests include hard-copy/softcopy appearance matching of images, simulation and modeling of both silver halide and digital imaging systems, development of color imaging algorithms, preferred color image reproduction, color restoration of faded and degraded images, gamut mapping, computational color science and development of color control tools and color management systems. He has authored numerous scientific papers in both the chemistry and color imaging fields in addition to a number of US patents.