Color Encodings: *sRGB* and Beyond

Kevin E. Spaulding
Eastman Kodak Company, Rochester, New York

Jack Holm
Hewlett-Packard Company, Palo Alto, California

Abstract

Most digital imaging products aimed at the open desktop market are designed to produce and accept image data in the *sRGB* color encoding. *sRGB* is defined with respect to the response of a reference CRT display. As a result, the image data can be easily interpreted, and can be directly displayed on a typical CRT without the need for additional color transformations. Thus, *sRGB* simplifies the workflow for softcopy-based viewing, editing, and image sharing. One intent of *sRGB* is to standardize the way in which images are stored and communicated in consumer digital imaging systems, thereby improving the interoperability of these systems. However, for non-CRT-centric applications, limitations associated with current *sRGB*-based workflows can negatively impact process complexity and image quality. This paper will discuss the pros and cons of several different approaches that have been proposed to overcome these limitations.

Color Gamut Considerations

sRGB Color Gamut Issues

One issue that is important in many digital imaging systems is the ability to fully and optimally utilize the color gamut of the output media. Because the *sRGB* color encoding is specified relative to the response of a standard CRT,[1] the colors that can be represented are limited to those within the color gamut of this standard display. As a result, storing images in *sRGB* can limit the capability of an imaging system to accurately reproduce colors outside the *sRGB* gamut. Film, scanners and digital cameras can both capture colors well beyond the *sRGB* color gamut. Therefore, it is necessary in *sRGB*-centric workflows to color render the captured image data into the *sRGB* gamut. Likewise, many output devices have color gamut shapes different from that of *sRGB*, with the device color gamut extending beyond the *sRGB* color gamut in some regions of color space, and the *sRGB* gamut extending beyond the device gamut in other regions. For example, as illustrated by the gamut slices in Fig. 1, photographic printers employed in digital photofinishing typically have a larger gamut for dark colors. And a smaller gamut for light colors. As a result, when printing *sRGB* images it is necessary to gamut map from the *sRGB* gamut to the actual output device gamut.

If the gamut mapping out of sRGB is designed to be complementary to the color rendering into sRGB, this practice can produce acceptable results. However, in many applications, the input and output processing will be independent, and therefore inconsistencies between proprietary color rendering and gamut mapping (or even gamut clipping) algorithms can often produce less than optimal results. Even in closed systems, where both the input and output processing can be coordinated, this co-optimization of the color rendering and gamut mapping can introduce a significant amount of additional computation and complication. These gamut restrictions are even more significant when color enhancements are applied to boost the colorfulness of an image, or where it is desired to accurately specify special colors (e.g., PANTONE* colors) that are outside the sRGB color gamut.

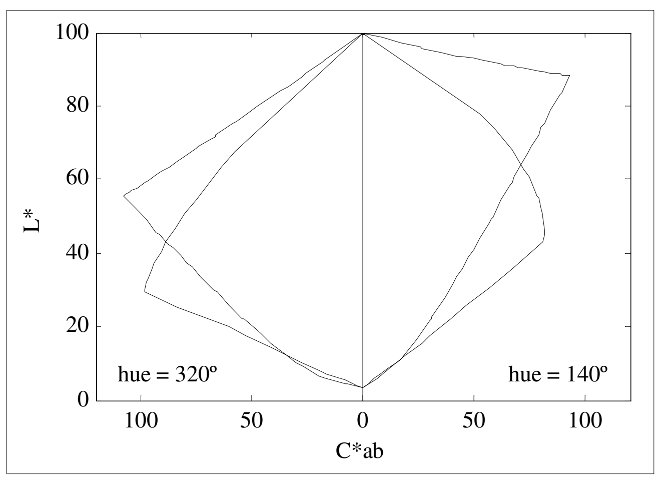

Figure 1. Comparison of CIELAB hue leaf slices through sRGB gamut (solid) and typical photographic print gamut (dashed). Magenta is to the left, and green is to the right. (The black-point luminance for the sRGB gamut has been adjusted to match that of the print media, as is common prior to gamut mapping.)

---

* PANTONE is a trademark of Pantone Inc.

Extended Color-Gamut Through YC_rC_b Color Encoding

Most current imaging devices and systems that produce *JPEG* files nominally encode color data according to the *sRGB* color encoding. However, as part of the *JPEG* compression process, these *sRGB* images are converted to a YC_rC_b color encoding prior to the actual discrete cosine transform (DCT) compression step to obtain a number of compression benefits. This is accomplished by simply applying a matrix transformation to the nonlinear *RGB* values to determine corresponding luminance-chrominance values. (The particular YC_rC_b color encoding that results from applying the standard *JPEG* color transformation matrix to *sRGB* color values is in the process of being standardized under the name(s) *sRGB YCC* or *sYCC*.[2,3,4])

This YC_rC_b color space has a substantially larger color gamut than the *sRGB* color space. Therefore, saturated colors that lie outside the *sRGB* gamut can be stored in a standard *JPEG* file by directly encoding the image data as YC_rC_b data, rather than first clipping the image to the *sRGB* gamut. One of the desirable features of this approach is that *JPEG* files containing extended-gamut YC_rC_b data will be backward compatible with existing *JPEG* file readers. Applications that know what to do with the extended-gamut image data can use it, while conventional file readers will simply clip off the extended-gamut information when the file is opened and use the *sRGB* image. The applications that retain the extended-gamut image data can enable output devices, such as photographic printers or inkjet printers, to make the best use of their color reproduction capabilities. In addition, the need for somewhat arbitrary gamut expansion on output is avoided.

The idea of using extended-gamut YC_rC_b image data is not new, although the full YC_rC_b gamut has not commonly been used with *JPEG* files. However, YC_rC_b color encodings have been used and standardized for many applications, and most *JPEG* compression software and hardware has supported the capability of dealing directly with YC_rC_b image data for many years. As a result, there should be no barrier to any digital camera/scanner creating and storing YC_rC_b images. Nor should there be a barrier for any printing system to make use of the extended-gamut YC_rC_b image data, regardless of the source.

Other Extended-Gamut Color Encodings

Recently, several other large gamut color encodings have been proposed. One such color encoding is *e-sRGB*.[3] This color encoding is an extended version of *sRGB* that allows the encoding of negative RGB values, as well as values larger than the *sRGB* white point. To support the extended range, the *e-sRGB* color encoding requires a minimum of 10-bits/pixel/color in digital precision. One desirable feature of this color encoding is that *sRGB* values correspond exactly to associated *e-sRGB* values, and can be computed using a simple 1-D LUT transformation. This feature allows *sRGB* images to be converted to *e-sRGB* and back without loss.

Another extended-gamut color encoding is *ROMM RGB*.[5] This color encoding achieves an extended color gamut by using a set of large gamut color primaries chosen to encompass the gamut of real world surface colors and to have a number of other desirable properties.[6] *ROMM RGB* is well-suited for both the storage of large-gamut digital images, as well as the application of common image manipulations such as tone scale modifications, color-balance adjustments, sharpening, etc. One desirable feature of *ROMM RGB* is that it can be used in workflows that are limited to 8-bits/pixel/color. The computation of an *sRGB* preview image from a *ROMM RGB* image is only slightly more complicated than for *e-sRGB*, requiring a LUT-matrix-LUT transformation. (Since there is not a one-to-one correspondence between *sRGB* code values and *ROMM RGB* code values, there may be a small loss of precision when going from *sRGB* to *ROMM RGB* and back, depending on the bit-depth.)

Since both the *e-sRGB* and *ROMM RGB* color encodings require that color transformations be applied in order to produce a video preview image, they are incompatible with many existing open system workflows. However, the recently approved *JPEG2000* image file format standard seamlessly enables the use of these extended gamut color encodings through the use of ICC profiles.[2,7] (The supported types of ICC profiles are restricted to the Monochrome Input and Three-Component Matrix-Based Input profile classes.) Until such time that support for the *JPEG2000* file format is widely implemented, it is anticipated that the use of *e-sRGB* and *ROMM RGB* will largely be limited to closed systems and higher-end color-managed applications. This implies that while these color encodings may be useful for the storage and manipulation of the digital images within a particular system, they will probably not be used for open image interchange in the short term.

All of the extended-gamut color encodings that have been mentioned thus far are *output-referred* in that they are intended to be representations of the color of a rendered output image. In some applications, *scene-referred* color encodings have been found to offer a number of desirable features. Scene-referred color encodings, such as the recently defined *RIMM RGB*,[6,8] are representations of the color of an original scene, and are generally designed to retain the full-dynamic range of the original scene capture. As a result, they offer the greatest flexibility for editing the image and specifying a preferred rendering of the scene. The use of scene-referred color encodings is only practical in open systems in conjunction with a file format like *JPEG 2000*, which supports the specification of a default rendering transform that can be used to form a corresponding output-referred image. This default rendering transform can be used by devices and applications that are not designed to work with images in a native scene-referred image state.

Image Consistency Considerations

One of the intentions in the standardization of *sRGB* was to improve the interoperability of digital images through the use of a common color image encoding. In practice, while there has been relatively widespread acceptance of *sRGB* as a common exchange metric in consumer digital imaging applications, *sRGB* images produced by different sources have been observed to vary significantly in their characteristics (e.g., overall brightness level, contrast, etc.). While some of these differences result from intentional variations in the color rendering aims used for different products, a significant portion of the variability can be attributed to the fact that the *sRGB* standard is somewhat ambiguous in several respects, and does not provide much useful guidance on how to color render an original scene into *sRGB*. This has left a lot of room for different companies to define their own "interpretations" of the *sRGB* standard.

One of the more significant sources of variation can be traced to ambiguity concerning the color rendering state of an *sRGB* image. The *sRGB* standard[1] states that "the encoding transformations between CIE 1931 XYZ values and 8-bit RGB values provide unambiguous methods for representing optimum image colorimetry when viewed on the reference display in the reference viewing conditions by the reference observer." This sentence implies that the *sRGB* image data is encoded in an *output-referred* image state that shall be interpreted as being optimally color rendered for the *sRGB* reference display. A somewhat different criteria is presented in Annex B of the standard, which proposes that this desired appearance can be produced by simply capturing a scene according to ITU-R BT.709.[9] Furthermore, Part 9 of the IEC 61966 series of standards[10] (of which the *sRGB* standard is Part 2) provides a method for characterizing a digital camera to produce scene colorimetry adjusted to a D_{65} white point, and Annex C recommends that this *scene-referred* colorimetry be considered *sRGB*.

These three conflicting options directly impact the creation and interpretation of *sRGB* image data. The colorimetry of an original scene is typically quite different than the colorimetry of a desirable output-referred image created from that scene.[6] As a result, given the ambiguity in the image state of an *sRGB* image, it is not surprising that there is substantial variation among implementations. The most visible manifestation of this variability will typically be differences in the overall image contrast since a boost in the luminance and chrominance contrast is usually an important feature of a well-designed color rendering function. *sRGB* images that are created by encoding scene-referred image colorimetry will generally be lower in contrast than those that are created from true output-referred colorimetry.

Another significant source of variability is differing interpretations of the adapted white luminance that should be assumed when viewing sRGB images. Some vendors have interpreted *sRGB* to be a *print-like* representation where a white with a code-value of 255 corresponds to a virtual white piece of paper. This is consistent with the paradigm that the computer's monitor is a virtual "desktop," and that a document window open on that desktop is analogous to a physical piece of paper on a physical desktop. In this scenario, prints made of a document that includes an *sRGB* image would typically map the white with a code-value of 255 to the output media white, and would map neutrals with code values less than 255 to proportionally darker neutrals on the print. This interpretation is also consistent with that assumed by most commonly available *sRGB* ICC profiles.[11]

Other vendors have interpreted the *sRGB* white point to correspond to an ideal *perfect white diffuser* with a reflectance of 100%. In this case, it is assumed that the viewer will interpret a white with a code value of 255 as being somewhat brighter than a piece of paper. Still other vendors have interpreted *sRGB* to be a *television-like* video representation, where the brightest colors are reserved to create the appearance of "whiter-than-white" specular highlights in the image. Images created using the perfect white diffuser paradigm will tend to be darker than those created using the print-like paradigm, and images created using the television-like paradigm will tend to be darker still.

The basic difference between these paradigms boils down to a difference in the assumption made about the adaptive white point of the observer who is viewing the image. This, in turn, reflects differing assumptions made regarding the typical workflow and image presentation. For example, if an *sRGB* image were to be pasted into a word processing document where the color of the page was a white with a code value of 255, then most viewers would interpret the brightness of the image relative to a code value of 255 being a typical piece of white paper. On the other hand, if the image were presented such that it filled the entire screen and all visual brightness cues such as title bars were eliminated, the viewer would largely adapt to the brightness level of the image. In this case, images created using the television-like paradigm would have the same capacity to create the appearance of specular highlights as they do on a conventional television. Appropriately controlling the image presentation can control the adapted white point of the observer almost arbitrarily. The specification of the image background and surround in the *sRGB* standard is not sufficient to clearly specify the adaptive white point luminance. This fact, combined with the ambiguity of the image state, has led different vendors to interpret it according to their view of an expected workflow.

Another source of variability is the specification of an unrealistic value for the *sRGB* veiling glare. The 0.2 cd/m^2 value specified in the standard is atypical for real CRT displays in the *sRGB* viewing environment, and CRT internal flare is not accounted for at all. The assumed viewer observed black point will affect the appropriate color rendering of an image, particularly in the shadows. By specifying an unrealistically low black point, implementers must choose whether to assume the specified value consistent with the standard, or a more realistic value that is consistent with workflows where the image data is sent without modification to real CRT displays.

Additionally, there appears to be a significant amount of variation in the way different vendors use and account for the reference viewing environment and display characteristics specified in the *sRGB* standard. Consequently, when creating a print from an *sRGB* image, some output paths adjust the image characteristics to account for differences in factors such as the overall luminance level, the viewing flare level and the image surround, while others neglect these differences. For example, aside from a chromatic adaptation step, the standard *sRGB* ICC profile, which is used in many color-managed printing paths, essentially neglects the differences between the ICC PCS reference medium/viewing environment and that associated with *sRGB*.[11]

To further complicate matters, it appears that some vendors may have ignored the *sRGB* specification altogether, and have labeled their images as "*sRGB*" even though they may have been rendered to a different RGB color encoding. For example, some vendors have historically created video RGB images for CRTs with a different gamma, white-point and/or RGB primaries than those specified for *sRGB*. In some cases, these vendors have apparently labeled their images as "*sRGB*" even though they are fundamentally inconsistent with the specification.

In a digital imaging workflow, any mismatch between the *sRGB* interpretation assumed in the capture process and the printing process can result in sub-optimal results for the final image. For example, if the image state, adaptive white point, viewer observed black point, or the gamma value assumed by the image source is different than that assumed by the output device, then the brightness and/or contrast of the printed image may be too low or too high. As a result, consumers may need to edit their images or manipulate printer driver settings in order to optimize results for a given print path and to achieve consistent results across a variety of devices. These same consistency problems will carry over into commercial digital photofinishing applications, and will apply to the YC_rC_b extension of *sRGB* as well.

There are a number of potential strategies that could be used to address these issues when making prints of *sRGB* images.

Assume Nominal *sRGB* Interpretation

The printing system can assume some nominal interpretation of *sRGB*. The interpretation assumed in any particular system may be designed to be consistent with a particular input source, or may be a compromise between some of the common input sources. This approach may be able to produce results that are acceptable for many input sources, but will not be optimal for any source that differs significantly from the assumed interpretation. The majority of current systems utilize this strategy.

Utilize Source-Dependent Processing

Most other approaches will increase system complexity. One alternative is to adjust the print path to account for the source-dependent image differences. For example, if all images from a certain vendor are known to print too dark, then images from that vendor can be lightened before they are printed. The source-dependent image differences can be characterized by the individual printing systems. Alternatively, the different image source vendors can be asked to provide guidance about image adjustments that are necessary to produce optimal results for a specific print path. This last concept is the basic approach used in Epson's PRINT Image Matching (PIM) technology[12] where digital camera vendors are asked to store metadata in the *JPEG* image file indicating how PIM-enabled print paths should alter the tone and color characteristics of the image so that it is "optimally rendered" for PIM-enabled output devices. The PIM metadata parameters include adjustments for image attributes such as brightness, contrast, color satura-tion, and color balance that can be used to account for differences in that camera's implementation of the *sRGB* standard.

While this approach can reduce the variability encountered with a specific output path, it can be seen that source-dependent processing will actually increase the global interoperability problem, and fundamentally breaks the "what-you-see-is-what-you-get" (WYSIWYG) paradigm that consumers have come to expect. For example, an image from one camera vendor may look darker than an image from another camera vendor when viewed on the user's CRT display, but when they are printed on a PIM-enabled printer they might have a similar density level. Then, if these same images were then sent to an Internet print fulfillment service, they might come out darker and lighter again. Consequently, the results that are obtained will not only be inconsistent between different input devices, but now they will also be inconsistent between different output devices.

Related to this problem is the fact that any image editing that a user performs on an image may be confounded with the image adjustments applied as part of the source-dependent processing. For example, if a consumer sees that an image from a particular camera is a little dark, he/she may edit the image to lighten it accordingly. Depending on whether or not the image editing application preserves the metadata, the edited image may get lightened again when the image is printed. Conversely, if the consumer uses the image with an application that does not preserve the metadata, this would effectively disable the metadata-driven processing. For example, if the user were to paste the image into a document, or even add text and save the image to a new file using an application that doesn't retain the image metadata, then the desired results would not be obtained even with a print path that is enabled to perform source-dependent processing.

Fundamentally, this approach does not address the root cause of the inconsistency in prints made from digital image files. Rather, it actually legitimizes these differences by sending the message to camera vendors "you can leave your images just the way they are and we will fix them later." To promote interoperability of digital images, it is critical to ensure that metadata specifying modifications to image appearance only be used in situations where it can be guaranteed that it will be properly interpreted by *all* applications and output paths. Even if there were industry-wide agreement that the use of this type of metadata were appropriate and should be standardized, it would be impractical to revise existing file format specifications such as *Exif* to require the use of this metadata because there is such a large installed base of applications and print paths that it would not be possible to update them all in a timely manner. The recently approved *JPEG2000* file format standard does provide a mechanism for supporting this feature through the required support for restricted ICC profiles.[2]

Utilize Image-Dependent Processing

Another approach that can be used to deal with the source-dependent image variability is to individually adjust images as they come through the printing system. This solution allows for the correction of image-dependent problems such as exposure errors, as well as the source-dependent variations associated with the different interpretations of *sRGB*. A brute force implementation of this strategy would be to have an operator manually adjust each image using a calibrated softcopy display. However, this would be time consuming and costly in any sort of high-volume workflow. Alternatively, automatic image processing algorithms can be used to estimate and compensate for the characteristics of individual images. The risk of this approach is that automatic algorithms can sometimes be fooled by the image content. For example, is an image dark because the exposure was incorrect, or because it is a picture of a black cat in a coal bin?

Reduce Variability In *sRGB* Interpretation

The ideal solution to the source-dependent variability problem is to minimize it by developing an industry consensus regarding the interpretation of *sRGB*/*sYCC* (or any other color encoding metric that may become popular in the future). While a detailed recommendation for a common interpretation is beyond the scope of this paper, several issues that should be addressed would include agreement on the interpretation of the *sRGB/sYCC* ambiguities mentioned previously, and the documentation of a default/reference means for mapping an original scene onto the reference display. (It is recognized that different camera vendors will want to implement their own preferred rendering aims, and therefore it would be inappropriate to force a single “look” onto all cameras. However, the definition of a reference “color rendering function” should reduce the unintended variability by providing a common baseline position.)

Conclusions

This paper has described two important issues associated with the use of consumer *sRGB* images in digital imaging workflows. The first issue is related to the color gamut of *sRGB*, and the resulting implications for accurately reproducing colors outside of the *sRGB* gamut. One desirable solution to this problem, which retains an extended color gamut while preserving backward compatibility and interoperability, is the direct use of *sRGB* YCC image data in *JPEG* image files.

The second issue that was discussed related to inconsistencies in the interpretation and implementation of the *sRGB* specification. While source-dependent processing can be used to address this inconsistency, this approach can actually result in more interoperability problems rather than less. A preferred solution is to work toward a common interpretation of *sRGB* throughout the digital imaging industry.

References

1. “Multimedia Systems and Equipment – Colour Measurement and Management – Part 2-1: Colour Management – Default RGB Colour Space – sRGB,” IEC 61966-2-1 (1999).
2. “Information Technology – JPEG 2000 Image Coding System – Core Coding System, Amendment 2, Inclusion of additional color-space,” ISO/IEC JTC 1/SC 29/WG1 N2223R, 10 July 2001.
3. “Photography – Electronic still picture imaging – Extended sRGB color encoding – e-sRGB,” PIMA 7667:2001.
4. Committee Draft for Vote of “Amendment 1 to Multimedia Systems and Equipment - Colour Measurement and Management – Part 2-1: Colour Management – Default RGB Colour Space – sRGB,” IEC 61966-2-1/A1/ Ed.1, 2001.
5. “Photography – Electronic still picture imaging – Reference Output Medium Metric RGB Color encoding: ROMM-RGB,” PIMA 7666:2001.
6. K. E. Spaulding, G. J. Woolfe and E. J. Giorgianni , “Optimized Extended Gamut Color Encodings for Scene-Referred and Output-Referred Image States,” J. Imaging Sci. Technol. **45**, 418-426 (2001).
7. C. Christopoulos, et al., “The JPEG2000 Still Image Coding System: An Overview”, IEEE Trans. Consumer Electronics **46**, 1103-1128 (2000).
8. “Photography – Electronic still picture imaging – Reference Input Medium Metric RGB Color encoding: RIMM-RGB,” PIMA 7466, Draft Version 1.0 (2001).
9. “Basic parameter values for the HDTV standard for the studio and for international programme exchange,” Recommendation ITU-R BT.709 (formerly CCIR Recommendation 709).
10. “Multimedia Systems and Equipment – Colour Measurement and Management – Part 9: Digital Cameras,” IEC 61966-9 (2000).
11. M. Nielsen and M. Stokes, “The creation of the sRGB ICC profile,” in *Sixth Color Imaging Conference: Color Science, Systems and Applications,* 253-257 (1998).
12. “Better prints from digital still cameras with PRINT Image Matching,’ white paper available at http://www. printimagematching.com (2001).

Biography

Kevin Spaulding received a BS in Imaging Science from Rochester Institute of Technology in 1983, and MS and Ph.D. degrees in Optical Engineering from the University of Rochester in 1988 and 1992, respectively. He has been with Eastman Kodak Company since 1983 where he is currently a Research Associate in the Imaging Science Division. He is also Technical Secretary for the CIE TC8-05 committee, which is tasked with defining standards for the unambiguous communication of color information in images. His research interests include digital color encoding, color reproduction, digital halftoning, image quality metrics, and image processing algorithms for digital camera and printers.