Computational Accuracy of RGB Encoding Standards

Henry R. Kang
Peerless Systems Corporation
El Segundo, California

Abstract

In this paper, we reported the computational accuracy of various RGB encoding standards using a set of color patches printed by an inkjet printer. These patches were measured in CIELAB under D_{65} illuminant. The color spaces under study were Adobe RGB98, Bruce RGB, an extended RGB, CIE 1931 RGB, CIE 1964 RGB, a proposed inkjet RGB, Kodak RIMM/ROMM RGB, a proposed Laser RGB, NTSC RGB, Photoshop wide gamut RGB, sRGB, sRGB64, and SMPTE RGB. The computational path was from L*, a*, and b* via CIEXYZ to RGB. The conversion from CIEXYZ to a specified RGB space followed the definition of that space. The resulting RGB values were scaled to integers in various bit-depth. For comparison purpose, the scaled integer RGB values were converted back to CIELAB. The color difference between measured and reversed CIELAB values was used as the measure for computational accuracy.

From this exercise, we were able to pinpoint major causes of the computational error. The remedies to the problem were recommended and preferred RGB color spaces were suggested.

Introduction

Color consistency based on colorimetric equivalence at the system level requires a device-independent color encod-ing standard, properly characterized color-imaging devices, and color conversion engines.[1] At the system level, one would encounter all kinds of input color representations and many different output specifications. If a device-independent color representation is selected as the intermediate exchange standard, we have the benefit of reducing the system complexity. Many colorimetric spaces such as CIEXYZ and RGB encoding standards can be used for this purpose. Color encoding standard provides format and ranges for representing and manipulating color quantities. An improper encoding scheme can severely damage the color conversion accuracy.

Experimental Design

The computational path was from CIELAB via CIEXYZ to RGB as shown in Fig. 1. If different white points were used for the source and destination spaces, CIELAB values were chromatically adapted to the destination white-point, then transformed to the RGB space using the matrix and vector multiplication. The conversion from CIEXYZ to a specified RGB space followed the definition of that space. Resulting RGB values were corrected via a gamma function, if it were defined, then scaled to integers in various bit-depth. For comparison purpose, the scaled integer RGB values were converted back to CIELAB as shown in Fig. 2. A given RGB input was scaled to the range [0, 1], adjusted by a nonlinear gamma correction, then transformed to CIELAB via CIEXYZ and chromatic adaptation. The color difference between measured and reversed CIELAB values was used as the measure for computational accuracy.

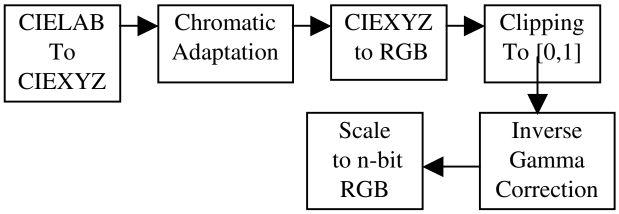

Fig. 1. Color transformations from CIELAB to RGB via CIEXYZ.

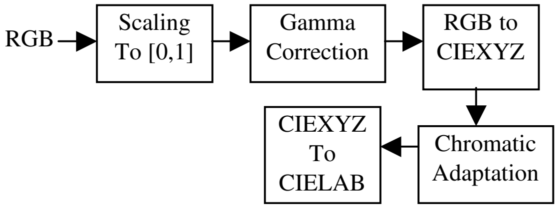

Fig. 2. Color transformations form RGB to CIELAB via CIEXYZ.

In this study, we compared the gamut size, locations of primaries, gamma correction (if specified), bit-depth of integer representation, and computational accuracy of the following RGB spaces: Adobe RGB98 (aRGB),[2] Bruce RGB (bRGB),[3] CIE 1931 RGB (cRGB31),[4,5] CIE 1964 RGB (cRGB64),[4,5] Extended RGB (eRGB),[6] Inkjet RGB (iRGB),[6] Kodak ROM RGB (kRGB),[7] Laser RGB (lRGB),[8] SMPTE RGB (mRGB),[9,10] Kodak RIMM/ROMM RGB (rRGB),[11-13] sRGB, sRGB64,[14] NTSC RGB (tRGB),[10] and Photoshop

Wide-Gamut RGB (wRGB).[2] The primaries of these color spaces were given in Table 1.

Table 1. Chromaticity coordinates of RGB color spaces.

Color space				White point	
		X	y	x	y
aRGB	Red	0.64	0.33	0.3127	0.3291
	Green	0.21	0.71		
	Blue	0.15	0.06		
bRGB	Red	0.64	0.33		
	Green	0.28	0.65		
	Blue	0.15	0.06		
cRGB31	Red	0.7347	0.2653		
	Green	0.2737	0.7174		
	Blue	0.1665	0.0089		
cRGB64	Red	0.7232	0.2768		
	Green	0.1248	0.8216		
	Blue	0.1616	0.0134		
eRGB	Red	0.701	0.299		
	Green	0.170	0.796		
	Blue	0.131	0.146		
iRGB	Red	0.70	0.30		
	Green	0.25	0.72		
	Blue	0.13	0.05		
kRGB	Red	0.873	0.144	0.3457	0.3585
	Green	0.175	0.927		
	Blue	0.085	0.0001		
lRGB	Red	0.7117	0.2882		
	Green	0.0328	0.8029		
	Blue	0.1632	0.0119		
mRGB	Red	0.630	0.340	0.3457	0.3585
	Green	0.310	0.595		
	Blue	0.155	0.070		
rRGB	Red	0.7347	0.2653		
	Green	0.1596	0.8404		
	Blue	0.0366	0.0001		
sRGB	Red	0.64	0.33	0.3127	0.3291
	Green	0.30	0.60		
	Blue	0.15	0.06		
tRGB	Red	0.67	0.33		
	Green	0.21	0.71		
	Blue	0.14	0.08		
wRGB	Red	0.7347	0.2653	0.3127	0.3291
	Green	0.1152	0.8264		
	Blue	0.1566	0.0177		

Coefficients for the decoding transfer matrix (from a chromatic RGB space to CIEXYZ) were computed from the chromaticity coordinates of primaries and the tristimulus values of the white point. The computation procedure can be found in Reference 10.

This experiment used 150 measured CIELAB values from a printed color test target by an Inkjet printer under D_{65} illuminant. The test target contained 125 color patches from 5-level CMY combinations and 25 additional three-color mixtures. This set of LAB values was converted to each RGB color space. For evaluating the problem of the color encoding, resulting RGB values were converted back to LAB values. Then, computational accuracy was judged by calculating ΔE_{ab} value between the input LAB and reversed LAB. If input data and RGB encoding were both under D_{65}, the chromatic adaptation was not performed. If the viewing conditions were different, the chromatic adaptation was just one matrix transform away if a simple van Kries model were used.

Results and Discussions

Using the sRGB encoding, we found that 38 out of 150 color patches required the clipping to put the encoded value within the range of [0, 1], the locations of these 38 colors spread across the color spectrum. This was an alarming 25.3% population of out-of-range colors. Most clipped sRGB points gave an error greater than 2 ΔE_{ab} with a maximum difference of 28.26 ΔE_{ab} and an average error of 2.22 ΔE_{ab}. Because of the clipping, these 38 points could not be reversed back to their original LAB values. This indicated that the sRGB color gamut was too small for a typical Inkjet printing. SMPTE RGB with a slightly smaller color gamut gave 45 out-of-range colors (30%). The maximum error was 27.42 and the average error was 2.92. Bruse RGB, having a larger color gamut, gave 27 out-of-range colors (18%) with a maximum error of 19.31 and an average error of 1.43. Similar to sRGB, SMPTE RGB and Bruse RGB spaces produced out-of-range colors that were scattered all over the color spectrum. Adobe RGB98 was an enlarged Bruce RGB; the red and blue primaries were the same as bRGB with the green primary at a higher chroma. The enlarged green region reduced the number of out-of-range colors to 15 (10%) by eliminating the clipping error in the green region, leaving out-of-range colors in the yellow, red, and purple regions. The maximum error was 19.83 and the average error was 1.17. Using the NTSC RGB (tRGB) primaries to encode the data, we found that there were 9 out-of-range colors (6%). These colors were clustered in the red region with one exception in yellow. The maximum error was 12.19 and the average error was 0.87. The tRGB primaries, having extended space in the green region, eliminated the green problem shown in sRGB and SMPTE RGB encodings. But, the x chromaticity coordinate of the green primary was too small such that it reduced the space in the red-magenta region. This caused many red colors to become out of range.

Starkweather suggested an extended RGB color space using primaries of three lasers with Helium-Neon Laser at λ=633 nm for red, Argon Laser at λ=514 nm for green, and Helium-Cadmium Laser at λ=442 nm for blue.[8] These spectral primaries gave a much larger color gamut, but we still obtained 8 out-of-range colors (5.3%) in green and yellow regions, indicating the wavelength of the green primary was too low. For 8-bit representation, the average error was 1.05 and maximum error was 28.26. Another spectral primaries, CIE 1931 RGB, gave 11 out-of-range colors (7.3%) in blue and green regions. This was because the green primary at λ=546.1 nm was too high in wavelength that reduced the green-blue gamut. The maximum error was 18.08 and the average error was 1.08. Other spectral primaries faired better, Adobe Photoshop® wide-gamut RGB gave only one out-of-range color - the most saturated yellow. The maximum error was 3.53 and the average error was 0.67. CIE 1964 RGB also gave the same out-of-range yellow with a maximum error of 3.68 and an average error of 0.64.

These results reveal that out-of-range colors produce the biggest color errors. They can't be brought into the range by extending the number of bits for encoding sRGB. As far as I know, there are two ways of dealing with this problem: the first one is to remove the clipping (Note that this is the approach used in sRGB64) and the second one is to enlarge color gamut. Using the first method, any out-of-range color is represented by either a negative value or a value greater than the maximum tone range. Negative values increase the complexity of the digital implementation. For example, one may not be able to use simple lookup tables. Another problem is that most electronic color devices can't render negative values. Therefore, some kind of mapping or clipping must be done. If a mapping is used, the color fidelity will be in jeopardy. If a clipping is used, the same computa-tional inaccuracy will resurface. Kress pointed out that any truncation of negative values will cause loss of image detail in highly saturated colors and color shifts. Monitor RGB such as sRGB and SMPTE RGB has a very small color space, much smaller than the gamut of photographic films and color hard-copy printers. Using monitor RGB without properly handling gamut mismatches, one would encounter objectionable elimination of color detail and shifts in hue, saturation, and lightness.[15]

A real solution is the second method of extending color gamut. As we have shown in various RGB encodings, the color gamut is controlled by the locations of primaries. The number, error magnitude, and location of the out-of-range colors are dependent on the size and shape of the color space. Therefore, we can eliminate out-of-range colors by expanding RGB color gamut with properly selected primaries; examples are iRGB and eRGB spaces. Primaries of the iRGB were chosen to encompass all 150 experimental data. As expected, there was no out-of-range color. In 8-bit depth, the average error was 0.58 and the maximum error was 2.94. Further accuracy improvements could be realized by increasing the bit-depth for encoding integer RGB. The average error became smaller and the error distribution became narrower as the bit-depth increased. There was practically no visually detectable error at 12 bits or higher. This was not the case for sRGB; the errors of the out-of-range colors remained the same regardless of the bit-depth. Moreover, the average error improved very slowly because error was contributed mostly from those out-of-range colors.

To accommodate wider applications in the system level, Kress proposed a spectral RGB space using primaries at 620, 530, and 460 nm. With this space, most of color gamut from photographic input materials and printer outputs could be encompassed.[15] Kang proposed a similar spectral eRGB space using primaries at 625, 532, and 467 nm.[6] This space was big enough to encompass the commercial Scanner/RGB, Monitor/RGB, Duoproof RGB, Inkjet CMYK, Printing offset press CMYK, and Hexachrome offset press shown in an Agfa literature.[16] Two even bigger RGB spaces, RIMM/ ROMM RGB and ROM RGB, were proposed by Kodak. These color spaces used primaries that were outside of the spectral locus (see Table 1). Most, if not all, real-world producible colors were enclosed within the gamut of these primaries.

Kress has compared the gamut of monitors (RGB709, P22, NTSC, and SMPTE), photographic films (Agfa, Fuji Photo, Kodak, and Konica), and printer paper outputs (SWOP, wax thermal transfer, dye diffusion, Kodak Q60, and graphic arts proofing material). He concluded that there was no single encoding scheme which resulted in minimal computation time, absence of image artifacts, device independence, and optimal quantization.[15] Süsstrunk, Buckley, and Swen derived a similar conclusion that no one RGB space was ideal for archiving, communicating, compressing, and viewing of color images.[17] The correct color space depends on the application. They recommended that if the desired rendering intent is known, the use of a wide-gamut space is the best choice for the situation that more than one type of output is desired. This situation describes the system environment that has various input and output devices. Their recommendation is in agreement with our finding that a wide gamut RGB is required in the system environment. The problems caused by using wide gamut RGB are relatively minor when compared to the problems caused by clipping and negative values. For example, the less numerical resolution can be overcome by using higher bit-depth and proper gamma correction. And, the mismatch between RGB encoding standard and real phosphor chromaticities becomes a true color gamut mapping that can be taken care of by a proper device characterization. In the system level color management, the device characterization is a must-have.

Conclusion

For the concern of the color reproduction in the system level, there are many problems such as the color gamut mismatch, color conversion technique, gray component replacement, quantization, resolution conversion, spatial scaling, halftoning, device characteristics, compression/ decompression, measurement error, and computational error. Among them, the color gamut mismatch is the most difficult one to deal with and perhaps gives the biggest color error. There are two kinds of color gamut mismatch: One stems from the physical limitation of imaging devices; for example, a monitor gamut does not match a print gamut; another one is due to the color encoding standard such as sRGB or SMPTE RGB. Color encoding standard is a man-made constraint to describe and manipulate color data. Considering many problems in the color reproduction, I believe that the color encoding should not be one. A result of this study shows that the color error induced by the improper color encoding can be eliminated. We should make every effort to eliminate the color error caused by the color encoding standard.

Moreover, to accommodate different color applications, we should make gamma correction an option because scanner/RGB does not need it.

References

1. H. R. Kang, *Color Technology for Electronic Imaging Devices*, SPIE press, Bellingham, Washington, Chapter 10, pp: 261-271 (1997).
2. Adobe® Photoshop 5.0.
3. B. Fraser and D. Blatner, http://www.pixelboyz.com.
4. CIE, Recommendations on uniform color spaces, color-difference equations and psychometric color terms, Supplement No. 2 to Colorimetry, publication No. 15. Paris: Bureau Central de la CIE (1978).
5. C. J. Bartleson, *Colorimetry*, in *Optical Radiation Measurements*, Vol. 2, p.43, p.47, p.86, Ed. F. Grum and C. J. Bartleson, Academic Press (1980).
6. H. R. Kang, Implementation issues of digital color imaging, Color Image Science Conference, CIS2000, Derby, England, April 10-12, (2000).
7. K. E. Spaulding, B. Pridham, and E. J. Giorgianni, Definition of standard reference output medium RGB color space (ROM RGB) for best practice tone/color processing, Kodak Imaging Science Best Practice Document.
8. G. Starkweather, Colorspace interchange using sRGB, http://www.color.org (1998).
9. SMPTE RP 145-1999, SMPTE C color monitor colorimetry, Society of Motion Picture and Television Engineers, 595 W. Hartsdale Ave., White Plains, NY 10607 (1999).
10. Xerox Corporation, *Color Encoding Standard*, Xerox Systems Institute, 475 Oakmead Parkway, Sunnyvale, CA 94086, p. C-3 (1989).
11. Eastman Kodak Company, Reference output medium metric RGB color space (ROMM RGB) white paper, http://www.colour.org/tc8-05.
12. Eastman Kodak Company, Reference input medium metric RGB color space (RIMM RGB) white paper, http://www.pima.net/standards/iso/tc42/wq18.
13. K. E. Spaulding, G. J. Woolfe, and E. J. Giorgianni, Reference input/output medium metric RGB color encodings (RIMM/ ROMM RGB), PICS 2000 Conference, Portland, Oregon, March 26-29, 2000.
14. M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta, A standard default color space for the Internet – sRGB, Version 1.10, Nov. 5 (1996).
15. W. C. Kress, First considerations of default RGB database, Proceeding TAGA, pp: 67-84 (1993).
16. *The Secrets of Color Management*, Digital Color Prepress Vol. 5, p. 9, Agfa Educational Publishing, P. O. Box 787, Randolph, MA 02368-0787 (1997).
17. S. Süsstrunk, R. Buckley, and S. Swen, Standard RGB color spaces, Proceeding of The Seventh Color Imaging Conference, pp. 127-134 (1999).

Biography

Henry R. Kang received his Ph.D. in physical chemistry from Indiana University and his MS. in computer science from RIT. He has worked on the ink jet ink analysis at Mead Corporation (1978-1980), ink jet ink formulation at Xerox Corporation (1980-1986), and digital color imaging at Xerox (1986-1998) and Peerless System Corporation (1998-2000). Currently with Aetas Technology Inc., he is managing the digital color science and technology development.

Henry is a member of the IS&T and SPIE. He is the author of two books, six patents, an article in IEEE Encyclopedia, and more than 20 papers in journals and proceedings.