

Color Conversion between sRGB and Internet FAX Standards

Henry R. Kang
Peerless Systems Corporation
El Segundo, California

Abstract

Color conversion between two encoding standards, sRGB and Internet FAX, is important for interchangeability and compatibility among Internet peripheral devices such as the monitor, printer, and FAX machine. To enhance the compatibility, we need to have the capability of converting between these color spaces accurately and cost effectively. To this end, we proposed a modular architecture for implementing this conversion via an internal exchange standard CIEXYZ.

The implementation was simulated in software using floating-point and integer computations. Several integer implementations using lookup tables (LUT) were presented. The computational accuracy of integer implementations was examined and the comparison was made with respect to the floating-point computation. Finally, the optimal imaging path and bit-depth were recommended.

Introduction

For Internet and FAX applications, we need to deal with two color encoding standards, Internet FAX color standard CIELAB and Internet default color standard sRGB. To enhance the compatibility, we need to have the capability of converting between these color spaces accurately and cost effectively. The general approach of the color conversion in the system level is to have an internal exchange standard. For this system, we select CIEXYZ as the internal standard because it is a common thread between the two standards, sRGB and CIELAB. Moreover, it is convenient for performing chromatic adaptation. The schematic diagram for CIELAB→sRGB transform is given in Fig. 1 and the inverse transform is given in Fig. 2.[1] Each module in Figs. 1 and 2 is implemented according to its definition and formulas.[2-7]

Implementation

From the color architecture given in Figs. 1 and 2, the system consists of four transforms: CIELAB→CIEXYZ, CIEXYZ→sRGB, sRGB→CIEXYZ, and CIEXYZ→ CIELAB. Each transform was implemented in software for floating-point and integer computations and they were tested using experimental data as well as synthetic values.

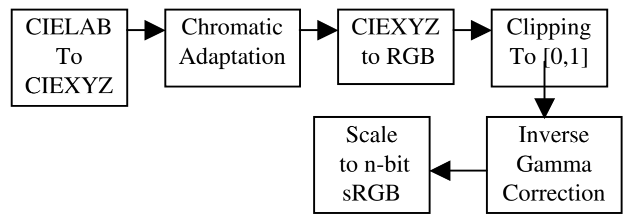

Fig. 1. Color transformations from CIELAB to sRGB via CIEXYZ.

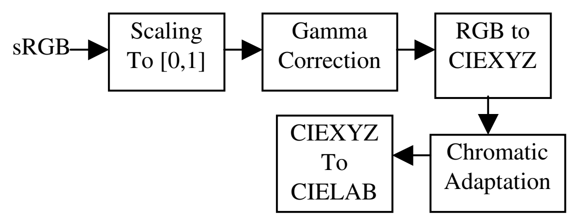

Fig. 2. Color transformations form sRGB to CIELAB via CIEXYZ.

CIELAB to CIEXYZ Transform

The CIELAB→CIEXYZ transform was implemented in two stages of table lookups. The first stage used three 1D lookup tables (LUT) with 256 entries to convert 8-bit ITULAB values to scaled L^*, a^*, and b^*.[2,3] One stored the results of $[(L^*+16)/116\times f_s]$, another for $(a^*/500\times f_s)$, and the third for $(b^*/200\times f_s)$, where f_s was the scaling factor. A scaling factor was used because this was a common approach to increase the accuracy of the integer arithmetic. The input values were multiplied by a scaling factor f_s (usually 2^n-1), performed the required integer computations, then scaled back to the original depth by dividing the same scaling factor.

The output of L^*-LUT was used to obtain Y. Outputs of a^*-LUT and L^*-LUT were added, the sum was used to obtain X. The output of b^*-LUT was subtracted from the output of L^*-LUT, the result was used to obtain Z.

The second stage had two implementations. The first approach used one 1D-LUT to store the computed cubic function. The output of this LUT was multiplied with the corresponding tristimulus value of the white point to give the resulting tristimulus value. For integer implementation, the white-point tristimulus values were scaled to the desired bit-depth. The second implementation folded the white-point multiplication into the LUT. Because X_n, Y_n, and Z_n were different, one needed three LUTs for three components. Again, the contents of these tables could be scaled for finding optimal bit-depth.

CIEXYZ to sRGB Transform

The white-point of the Internet FAX color standard is D_{50}, where the white-point of the sRGB is D_{65}. Therefore, a chromatic adaptation is needed for this transform. If a simple von Kries model is used,[5] the chromatic adaptation and matrix transform can be combined and implemented as an integer matrix multiplication using scaled coefficients. This implementation gave one table lookup, nine multiplications, six additions, and, optionally, three divisions for scaling back to n-bit. Resulting values were clipped to [0,1] in floating-point computation or [0, maxLevel] in integer arithmetic.

The last step in CIEXYZ→sRGB transformation was the gamma correction. For integer arithmetic, the gamma correction was implemented as a 1D-LUT using integer sRGB value as the index to the table.

sRGB to CIEXYZ Transform

The sRGB→CIEXYZ conversion was implemented in two different ways, depending on the need for performance or cost. A low-performance and low-cost implementation used a 1D-LUT to store the computed results of the scaling and gamma correction, followed by a matrix multiplication. The content of the table could be integer or floating-point values in which the integer representation had lower computational cost and lower accuracy but required few memories for storage. The bit-depth of the input sRGB determines memory requirement and computational accuracy because the number of LUT entries is equal to 2^n, where n is the input bit-depth.

The second step was to compute scaled XYZ from scaled RGB, followed by chromatic adaptation. Since they were linear transforms, the conversion and adaptation could be combined to give a single matrix that converted gamma-corrected sRGB to chromatic-adapted XYZ. To employ integer arithmetic, the coefficients were scaled to integer. This implementation gave one table lookup, nine multiplications, six additions, and, optionally, three divisions for scaling back to n-bit.

For high performance implementation, the sRGB to CIEXYZ transform was implemented in two stages of table lookups. The first stage was the same as the low-cost and low-performance implementation. But, the second stage was implemented as nine 1D LUTs to store products of each coefficient with a component of the scaled RGB. In this way, the whole computation became ten table lookups and six additions. The enhanced performance was at the expense of the memory cost. The memory requirement could become quite costly. For example, an 8-bit input and 12-bit depth require 512 bytes for the first LUT and 73,728 ($2\times9\times2^{12}$) bytes for the second stage of LUTs.

CIEXYZ to CIELAB Transform

The cubic-root function $f(t)$ of the CIEXYZ→CIELAB transform is bounded in the range of [16/116, 1] because t has the range of [0, 1]. If the input CIEXYZ is scaled and represented by 8-bit integers, the values of the cubic-root function can be pre-computed and stored in a 1D-LUT. If the scaling factor is not the same for XYZ components, we need three 1D-LUTs, one for each component of the white-point, to store the cubic-root values. Using LUTs, we reduced the computational cost and enhanced the performance by removing the run time computation of the cubic-root function. This implementation gave a total of three table-lookups, three additions/subtractions, and three multiplications. Further performance enhancement could be realized by using LUTs to store results of $116\times f(t)-16$, $200\times f(t)$, and $500\times f(t)$, respectively. This implementation gave five table-lookups and two subtractions. The difference was that two more LUTs were required, but the computa-tional cost was significantly reduced.

Results and Discussions

We performed software simulations by varying the bit-depth of LUTs and matrix coefficients of each individual transform for the purpose of finding the optimal bit-depth. For sRGB, the bit-depth n of the first LUT and the bit-depth m of second LUTs or matrix coefficients were varied independently. Computational accuracy was judged by comparing the integer result with the floating-point computation in ΔE_{ab} value. After all four modules were individually tested, we put them together in the sequence shown in Figs (1) and (2) for checking the overall performance with respect to the computational error.

CIELAB to CIEXYZ Transform

This experiment used 147 color patches produced by a xerographic copier, having 20 levels of CMY primaries, 15 levels of the black primary, 7 levels of M-Y, C-M, and C-Y two-color mixtures, and 51 three-color mixtures. Measured CIELAB values were converted to ITULAB, and used as inputs to the first LUT.[2,3]

Table 1 gave the average ΔE_{ab} of the low-cost implementation. For a given bit-depth of the second LUT, Table 1 indicated that the bit-depth of the first LUT had little effect on the computational accuracy. Accuracy improved as bit-depth of the second LUT, implementing the cubic function, increased. However, the improvement leveled off around 12 bits. The increasing of bit-depth not only improved the accuracy but also narrowed the error distribution.

Similar to the low-cost implementation, computational accuracy of the high-cost implementation improved as bit-depth of the second set of LUTs increased. It also leveled off around 12 bits.

Table 1. Average ΔE_{ab} for low-cost implementation of the CIELAB to CIEXYZ.

Bit-depth of First LUT	Bit-depth of second lookup table 7-bit	8-bit	9-bit	10-bit	12-bit	14-bit
8-bit	2.17	1.40	0.83	0.74	0.68	0.68
9-bit	2.09	1.37	0.83	0.72	0.62	0.61
10-bit	2.06	1.27	0.76	0.67	0.53	0.51
12-bit	---	1.32	0.79	0.62	0.53	0.53
14-bit	---	1.30	0.84	0.63	0.55	0.52

CIEXYZ to sRGB Transform

This experiment used 150 data points, having 125 color patches from 5-level combinations of a RGB cube and 25 three-color mixtures. The CIEXYZ values of these color patches were measured under D_{50}.

This transform gave the biggest error because the clipping was used in this stage. The errors of out-of-range colors, depending on the distance from the triangular gamut boundary, were usually big. For those in-gamut colors, using integer computation, the average difference between 8-bit CIEXYZ inputs and 8-bit reversed CIEXYZ (obtained from the next transform sRGB→CIEXYZ) was 1.7 digital counts with a maximum of three digital counts.

Without clipping, there was no computational error for the floating-point computation. For integer implementation without clipping, the average error was 3.5 counts for 8-bit representation of sRGB. The error decreased as the bit-depth increased. It leveled off around 12 bits at 1.1 counts.

sRGB to CIEXYZ Transform

The converted sRGB from previous transform was used as inputs for this transform. Table 2 gave the average ΔE_{ab} of 150 data points under various bit-depth combinations. For a given bit-depth of the first LUT, Table 2 showed a substantial accuracy improvement as bit-depth of the matrix coefficient increased from 8 to 9 bits. Not much improvement was gained after 9-bit. On the other hand, for a given bit-depth of matrix coefficients, accuracy improved as bit-depth of the first LUT increased. The improvement leveled off around 12 bits. The bit-depth increase not only improved the accuracy but also narrowed the error distribution. For all practical purposes, the 12-bit and 14-bit results were as good as the floating-point computation.

The two-stage LUT implementation showed a similar trend. The difference was the magnitude of errors in which the lookup approach had a higher computational accuracy.

Table 2. Average ΔE_{ab} of sRGB→CIEXYZ transform.

Bit-depth of First LUT	Bit-depth of matrix coefficients 8-bit	9-bit	10-bit	12-bit
8-bit	1.31	1.15	1.15	1.17
9-bit	0.82	0.55	0.59	0.56
10-bit	0.53	0.30	0.30	0.27
12-bit	0.42	0.19	0.12	0.08
14-bit	0.42	0.18	0.11	0.05

CIEXYZ to CIELAB Transform

In this transform, we used 125 sets of scaled CIEXYZ values. Five different scaling factors f_s with values 1 (no scaling), 2, 4, 8, and 16 were used. For the case of no scaling, a 93.6% of the total data gave a color difference less than 1 ΔE_{ab}. The percentages for other cases were 100% for f_s=2, 81.6% for f_s=4, 73.6% for f_s=8, and 70.4% for f_s=16. The scaling by a factor of 2 provided the most accuracy results with narrowest error distribution.

Combined Computational Error

Using 150 data points, we have examined the overall computational error from CIELAB via CIEXYZ with chromatic adaptation to sRGB, we then converted sRGB back to CIELAB. The difference between the initial CIELAB and inverted CIELAB was the measure of the computational accuracy.

Table 3. Average color differences of the overall conversion under various conditions.

Condition	Average Color Difference 8-bit	9-bit	10-bit	12-bit	14-bit
Float-point	2.39	2.20	2.12	2.05	2.04
Float-point/Gamma	2.22	2.12	2.08	2.04	2.04
Integer, 150 data	3.73	3.06	2.84	2.74	2.72
Integer, 112 data	1.93	1.12	0.85	0.74	0.74
Integer, 9-bit input	2.00	1.12	0.88	0.64	0.66
Integer, 10-bit input	1.95	1.14	0.83	0.66	0.67

For floating-point computations, we obtained average ΔE_{ab} values ranging form 2.04 to 2.39 (See Table 3), depending on the number of bits used to represent sRGB. The average value decreased as the bit-depth increased, but the improvement leveled off around 12 bits. The biggest error was 28.3 ΔE_{ab} units. The gamma correction reduced computational error somewhat.

For integer computations, we obtained average ΔE_{ab} values ranging form 2.72 to 3.73, corresponding to 4.01 to 5.75 in digital counts when represented in 8-bit ITULAB format (see Table 3). The maximum error remained about the same as the floating-point computation. By removing the out-of-range colors, we obtained much smaller errors, ranging from 0.74 to 1.93. On increasing the input bit-depth of the ITULAB, we didn't see any significant improvement in the computational accuracy. This result reconfirmed previous studies that the 8-bit depth is sufficient to represent the visually linear color spaces such as CIELAB.[8]

Conclusion

As a summary, this study revealed that the bit-depth of the input ITULAB had little effect on the computation accuracy. The computational accuracy improved as bit-depth of the cubic-function LUT increased, but the improvement leveled off around 12 bits. The increasing of bit-depth also narrowed the error distribution.

In the sRGB to CIEXYZ transform, the optimal bit-depth for the high-cost implementation was determined at 10 bits for the first LUT (8-bit and 9-bit were a little bit low in computational accuracy). This implementation gave 1024 entries for the second set of LUTs and required 18432 bytes (1024×9×2, if the bit-depth of the LUT element is greater than 8) to store all 9 coefficients. These numbers indicated that any bit-depth greater than 10 might be too costly to implement. The bit-depth of the second LUT had little effect on the computational accuracy; an 8-bit depth was fine. For low-cost implementation, we could use 12 or 14 bits for the first LUT because no LUT was used in the second stage. However, we recommend to encode matrix coefficients at 9 bits or higher (8-bit was a little bit low in accuracy).

References

1. H. R. Kang, Implementation issues of digital color imaging, Color Image Science Conference, CIS200, Derby, England, April 10-12, (2000).
2. L. McIntyre and S. Zilles, File format for Internet FAX, http://www.itu.int/itudoc/itu-t (1997).
3. ITU-T.42, Continuous tone color representation method for facsimile, available at http://www.itu.int/itudoc/itu-t (1996).
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. M. D. Fairchild, *Color Appearance Models*, Chapter 9, pp. 199-214, Addison-Wesley (1998).
6. M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta, A standard default color space for the Internet – sRGB, Version 1.10, Nov. 5 (1996).
7. IEC/3WD 61966-2.1: Colour measurement and management in multimedia systems and equipment – Part 2.1: Default RGB colour space – sRGB, http://www.srgb.com (1998).
8. L. Xu and R. Holub, Color spaces for image representation and image processing, Proc. TAGA, pp. 114-138 (1992).

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.