Color Transformation Accuracy and Efficiency in ICC Color Management

Huanzhao Zeng
Hewlett-Packard Company, Vancouver, Washington
Email: huan_zeng@hp.com

Mary Nielsen
Hewlett Packard, Boise, Idaho

Abstract

ICC color management provides many benefits for standardizing color management and for accurate color transformation. However, problems are existed, including the creation of ICC profiles, the implementation of CMM (color management module), and the specifications of the ICC format. The misinterpretation of ICC specifications results in inaccurate data representation in ICC profiles. The lack of CMM specification results in the interoperability among different CMMs and inaccuracy of color transformation in many CMMs. In this paper, we focus on following topics and provide some solutions to improve the color transformation accuracy and the color reproduction quality in ICC color management:

1) How to select a PCS (XYZ or L*a*b*) and a LutType (Lut8Type or Lut16Type) for profile creation;
2) How to determine the sizes of 1-D LUTs (lookup tables) and the size of the multidimensional LUT for accurate data representation and color transformation;
3) The importance of 1-D LUTs in Lut16Type tag;
4) Profile linking for CMM implementation; and
5) The implementation of a smart CMM for printing applications without using private tags in profiles.

1 Introduction

There are many different approaches to reproduce reliable and desirable colors as colors are transferred across a color processing system. One intuitive approach is to fix an input device and an output device and to characterize the system for the color transformation from the input color space to the output color space. This kind of color processing system is called a closed-loop system. Although easy to calibrate, it is problematic as the world moves to network oriented open architecture. As an open color system comprises unknown number of input and output color devices, a closed-loop approach is not capable of calibrating the system. If all device-dependent colors from different color devices are converted into a device-independent color space, it will be easier to maintain color specifications of different devices. A color management system based on a well-defined color system, such as CIE color spaces, as a connection bridge meets the requirement for network based color management. This kind of color management system and components was standardized by the International Color Consortium (ICC). The ICC profile format[1], which was evolved from the ColorSync 2.0 format, is served as a cross-platform device profile format for color device characterizations.

The profile connection space (PCS) is the heart of the ICC color management, which is CIE XYZ or CIE L*a*b* in D50 illuminant with specified viewing conditions. The color transformation by ICC color management is based on the Profile-PCS-Profile model. Any device color is communicated with other device colors through the PCS. The first step for applying the ICC color management is to create ICC profiles for each device, which is the device color calibration step. This step is critical for the quality of the color transformation of each device in a “smart profile and dumb CMM” model. Besides the algorithms to create ICC profiles, the ICC profile specification (e.g. tag formats, the PCS definition) also influences the color representation result.

Another factor that influences the color reproduction is the CMM implementation.[2] As a part of the ICC color management, a CMM is a component to interpret ICC profiles for color transformation. A CMM takes ICC profiles, links them, and performs color transformation. The implementation method and the interpolation accuracy decide the overall color conversion accuracy. Because of the lack of CMM specifications, different CMMs were implemented differently therefore the interoperability is not guaranteed.

ICC color management has been applied to a very wide range of markets.[3] Because different markets have different requirements, it is difficult to address all of its limits and provides relevant solutions to all problems. Based on the author’s experience, methods to increase the accuracy for ICC profile creation will be described, the impacts of implementing a CMM by different approaches will be discussed, and methods to implement CMM for accurate color transformation will be addressed. An approach to implement a "smart" CMM compatible with the current ICC profile specification will be introduced. A method to simplify the implementation to support both the version 2 and the version 4 profiles in a CMM will be presented.

2 Representing Data Accurately in ICC Profiles

There are many ICC profiling tools available in the market, such as ICC profiling tools from Agfa, Color Solutions, ColorSavvy, Pictographics, Heidelberg, GretagMacbeth, and Monaco Systems. For printer ICC profiling, some tools use the 16-bit Lut16Type, while others use the 8-bit Lut8Type. For the LutType (Lut8type and Lut16Type) tags, the sizes of 3-D or 4-D tables and the sizes of the 1-D tables in the Lut16Type tags created by different tools are different. Why are they different? Are they optimized? Since LutType is very important for printer ICC profiling, methods to represent LutType data in ICC profiles accurately will be discussed in the following sub-sections.

2.1 Selecting a LutType (8-bit or 16-bit LUT) for Profile Creation

The byte size of LutType data is about double by converting from 8-bit type to 16-bit type. For Lut8Type data, there are only 256 steps of tones. This is usually enough for representing digital documents. However, if color conversion is processed by more than one step and each step is represented in 8-bit depth, the final accuracy will be lower than 8-bit. In the Lut8Type structure, there are three parts of 8-bit tables, a set of 1-D input tables, a multidimensional LUT, and a set of 1-D output tables. If any of the 1-D input and output tables is not an identical table, the final accuracy may be lower than 8-bit. This is probably the reason that most Lut8Type data in ICC profiles have identity 1-D input and output tables.

Because linear interpolation methods are used for real-time multidimensional interpolation in CMM implementations, any nonlinear relationship between two grid points is approximated by linear transformation therefore interpolation error is unavoidable. To reduce the error, one should try to decouple the non-interdependent nonlinearities of the input and the output color spaces and to put them into the corresponding input 1-D tables and output 1-D tables. It will result in a more linear multidimensional LUT, thus improves the color transformation accuracy. If any input 1-D LUT or output 1-D LUT is not identical, LUT16Type should be used to guarantee at least 8-bit precision. Another benefit of using Lut16Type is that the multidimensional LUT is also represented in 16-bit so the accuracy is higher. An example to use 1-D LUTs to increase data representation accuracy in LutType tags is shown in the appendix.

In the next ICC specification (version 4.x), all 1-D tables in LutB2AType and LutA2BType data are represented in 16-bit precision or in parametric type, so the inaccurate 8-bit 1-D representation will not exist.

2.2 Selecting a Profile Connection Space (PCS)

Serving as the heart of the ICC architecture, PCS is the bridge to connect different color spaces. The current official PCS color spaces are CIE XYZ and CIE L*a*b*. CIE XYZ color space is not a uniform color space as is CIE L*a*b* color space. In this sense, CIE XYZ color space may not be as good as CIE L*a*b* color space for the color space sampling for linear interpolation, thus it may not be as good as CIE L*a*b* color space serving as PCS for LutType data representation. This is probably the reason that most printer ICC profiles use CIE L*a*b* as PCS if LutType data tags are used for color transformation. However, there are exceptions. CIE XYZ color space can be converted into a monitor RGB color space (without considering nonlinear gamma correction or TRC transformation) easily by a 3 by 3 matrix multiplication. This kind of linear relationship does not exist in the conversion between CIE L*a*b* and a monitor RGB color space. The close relationship between a monitor RGB color space and the CIE XYZ color space makes it possible to sample monitor RGB gamut surface points in the CIE XYZ grid points of the BToA0 tag (or BToA1 and BToA2 tags) in a printer profile. Thus those colors in the device gamut surface can be mapped more accurately, and interpolations using [monitor ICC profile] -> PCS -> [printer ICC profile] conversion can be performed more accurately. For example, it will be more accurate to convert a monitor's primary colors into a printer's primary colors. Another example is to use XYZ as PCS to create e-sRGB color space ICC profile, which will be described in the appendix.

2.3 The Effects of the Sizes of 1-D and Multidimensional LUTs

The LutType data is processed via the following sequence: [3 × 3 matrix] -> [1-D input LUTs] -> [multi-dimensional LUT] -> [1-D output LUTs] in the current ICC specification (version 2.x). While the entries of the input and the output 1-D LUTs of the Lut8Type tag are fixed to 256, they are between 2 and 4096 for the Lut16Type tag. Because a multidimensional LUT uses a lot of memory or disk space, the LUT size of a multidimensional LUT is usually no larger than 33.

The color transformation is usually more accurate with larger 1-D LUTs in Lut16Type data if the 1-D transform is not linear. If the 1-D LUTs are linear or close to linear and the sizes of 1-D LUTs are very large, it may not further improve the accuracy as the size of 1-D LUTs are further increased. Different CMM implementations also have impacts on the decision of 1-D LUT sizes. If a CMM simply takes the 1-D LUTs and use them for 1-D lookup in the linking step, we should create larger 1-D LUTs for higher accuracy. If a CMM takes the 1-D LUTs and interpolate them to create larger 1-D LUTs, or if a CMM applies interpolation using 1-D LUTs as sampling points in the profile linking process, using relatively smaller 1-D LUTs will also achieve about the same accuracy as using larger 1-D LUTs if the 1-D curves are not changed very dramatically. If the CMM implementation is not known, using large 1-D LUTs for nonlinear 1-D transformation guarantees the color transformation accuracy.

While creating a large multidimensional LUT helps the color transformation accuracy, it dramatically increases the profile size and the memory for profile linking. If decreasing the LUT size of the multidimensional LUT from about 33 to about 17, the high chroma gamut surface colors are usually desaturated (the chromas are decreased). The reason is that the multidimensional interpolation takes a few neighbor points for linear interpolation and these points have more different hue and/or lower chroma if the LUT size is smaller. If decreasing the size of the multidimensional LUT from about 17 to about 9, besides losing chroma for high saturated surface colors, non-smoothness may happen in near gray colors. If creating a small profile is critical, the input 1-D LUTs are places to redefine the grid points so that there are denser samplings in the important color areas.

If a profiling tool has problems in smoothing gamut surface colors, it can be improved by using a smaller multidimensional LUT. Although this is not a good solution, it could be used as a quick and easy solution for smoothing the gamut surface.

2.4 Gamut Mapping Problem for Output ICC Profiles

To transform colors from an input device to an output device for best visual result, gamut mapping4 is an essential step for compressing regions that the output device has a smaller sub-gamut, stretching regions that the output device has a larger sub-gamut, and sometimes rotating hues so that the input primaries match the output primaries5. This means that both the input and the output gamut information must be known. However, the flexibility of the current ICC color architecture makes this impossible in general, therefore it makes optimal gamut mapping impossible. Because the input gamut is unknown as an output ICC profile is created, the gamut mapping is usually performed by simple clipping, or by assuming a certain input gamut. Since the gamut mapping is mostly performed in output ICC profiles, this is mainly the problem for output ICC profiles. One example to show the problem is the color transformation from a monitor RGB color space into a printer CMYK color space for graphic objects, in which the monitor pure yellow is not guaranteed to map to the printer pure yellow. The gamut mapping problem is the major factor that prevents the application of the saturation rendering intent in ICC color management.

It seems that the only solution is to perform the gamut mapping at the time that both the input and the output devices are known, which is in the linking step. This means that the gamut mapping must be performed at real-time. However, this requires that a CMM implements gamut mapping algorithms that are robust and fast for real-time processing.

In order to improve gamut mapping result, some ICC profiling tools provide features to decide an input color space as an output printer ICC profile is created or edited. However, this kind of solution is contradicted to the open color architecture. A different solution will be described in Section 3.2.6.

3 The Implementation of ICC Color Management Module

An ICC color management module is a color engine for color transformation using ICC profiles. Since ICC profiles often use lookup tables to approximate color conversion relationships, the final values obtained by CMM interpolations may not be desirable. Because there is no CMM specification, different CMMs perform ICC linking and interpolations differently. Therefore different color transformation results may be obtained with the same ICC profiles but different CMMs. Following sub-sections are a few topics about different linking implementations and their impacts to the results.

3.1 Profile Connection Spaces

ICC color management is based on the Profile -> PCS -> Profile model. Because there are two different PCS color spaces, CIE XYZ or CIE L*a*b*, a CMM must perform the conversion from one PCS to the other if the PCS in the first profile is different from the PCS in the second profile. If the PCS is CIE L*a*b*, there are two different bit-depths: 8-bit and 16-bit. Thus, there are eight pairs of PCS to PCS transformation. If a CMM supports only one PCS (either XYZ or L*a*b*) and converts LutType tag that uses the other PCS into a new LutType tag that uses the supported PCS, color conversion result may be unexpected. Because the grid points of a multidimensional LUT for one PCS are different from those for the other, the interpolation results will be different if the PCS is changed from one to the other, especially for colors near or in the gamut surface. Therefore, supporting both XYZ PCS and L*a*b* PCS without converting multidimensional LUTs is very important for the CMM interpolation accuracy.

Our approach to reduce the combination for PCS to PCS conversion without sacrificing the accuracy is to convert any Lut8Type tag into Lut16Type tag before linking. The PCS to PCS conversion will be reduced from eight to following four possibilities: XYZ to 16-bit L*a*b*, XYZ to XYZ, 16-bit L*a*b* to XYZ, and 16-bit L*a*b* to 16-bit L*a*b*. In these four combinations, only two conversions are required.

3.2 Linking Two ICC Profiles for Color Transformation

For fast color transformation, a CMM typically links a set of ICC profiles for color transformation to form a linking object or a deviceLink profile for one-step or fewer steps of color transformation. The merging of some 1-D LUTs with multidimensional LUTs is usually unavoidable in the linking step. Since the multidimensional linear interpolation cannot reproduce the nonlinear operation between grid points, interpolation errors cannot be avoided if nonlinear 1-D curves are merged into the multidimensional LUT. To achieve high color accuracy, a CMM should try to delay the merging of 1-D LUTs with a multidimensional LUT in the linking process. Several linking approaches are described in the following sub-sections. While only two ICC profiles are used for the linking in the following discussions, these approaches can be applied to link more than two ICC profiles. We assume both ICC profiles use LutType data for color transformation in Sections 3.2.1 to 3.2.4. The application of color transformation using profiles created by colorant tags and TRC tags is discussed in Section 3.2.5. Although these linking processes are described based on the current ICC specification, they are still valid for implementing the next version (version 4.x) of the ICC specification.

3.2.1 Merging Each LutType Data Tag into a Single Multidimensional LUT

A CMM takes the AToBi (i = 0, 1, or 2) tag of the first ICC profile and the BToAi (i = 0, 1, or 2) tag of the second ICC profile for color transformation. A simple linking approach is to create a multidimensional LUT for each LutType tag by combining the sequence operations of [matrix] -> [1-D LUTs] -> [multidimensional LUTs] -> [1-D LUTs], then to merge two multidimensional LUTs to form a final LUT for the color transformation by multidimensional linear interpolation.

There are two sources of linking errors coming from this approach. Because the 1-D LUTs are merged with the multidimensional LUT for each ICC profile’s LutType data, the 1-D nonlinear factors are merged into the multidimensional LUT transform. This is the first source of linking error. The second source of the errors comes from the merging of two multidimensional LUTs. The accuracy of color transformation is poor by applying this kind of linking process.

3.2.2 Creating a Multidimensional LUT through Step-by-Step Sequence Operations

To reduce the linking error in the above approach, a final multidimensional LUT can be created by the step-by-step sequence operations through [matrix] -> [1-D LUTs] -> [multidimensional LUTs] -> [1-D LUTs] of both AToBi and BToAi tags. The operation through the 1-D LUTs can be performed by linear interpolation, or by simple 1-D lookup. If the 1-D lookup approach is applied, sufficiently large 1-D LUTs should be created to guarantee the accuracy.

Color transformation accuracy is improved substantially over the previous approach if 1-D LUTs are not linear.

3.2.3 Linking Two ICC Profiles to Form a DeviceLink Profile

Although the second approach improves the accuracy, it can be further improved by keeping a set of input 1-D LUTs in front of the multidimensional LUT and a set of the output 1-D LUTs behind the multidimensional LUT. This kind of object could be created as a deviceLink ICC profile without the 3 by 3 matrix conversion step (set the matrix to identity in the profile). Since a real-time color transformation with a step of 3 by 3 matrix conversion (nine 1-D lookup operations plus six additions/subtractions) slows down the process, a 3 by 3 matrix conversion step should not be used in this approach.

Applying the linking object created by this approach, the color transformation flow becomes [1-D LUTs] -> [multidimensional LUT] -> [1-D LUTs]. The color conversion is slower than using the previous approach, but the accuracy is improved if the input and/or the output 1-D LUTs are not linear. To speed up the color transformation, the input or the output 1-D LUTs could be merged with the multidimensional LUT, depending on the linearities of the input and the output 1-D LUTs.

3.2.4 Color Transformation without Linking

In the above approach, the output 1-D LUTs of the first ICC profile and the input 1-D LUTs of the second ICC profile are merged with two multidimensional LUTs to form a multidimensional LUT (including the merging of matrixes). To avoid any nonlinear 1-D LUTs merged with multidimensional LUTs, a color transformation could be performed by the step-by-step sequence operations of [matrix] -> [1-D LUTs] -> [multidimensional LUT] -> [1-D LUTs] through all ICC profiles. It avoids the nonlinear factors in 1-D LUTs merged with the multidimensional LUTs. Since real-time color transformation is almost always performed by integer operations, so many steps of integer operations may lower the final accuracy. Therefore, high bit-depth operations are required for desired accuracies. Because the color transformation goes through so many steps, it is very low efficient if several ICC profiles are linked together. The floating-point computation of this approach can be used for debugging and for high accurate conversion, such as for profile re-generation and profile editing, and for research.

3.2.5 Linking ICC Profiles Defined by Colorant Tags and TRC Tags

A monitor ICC profile is typically comprised by red, green, and blue colorant tags, and red, green, and blue TRC tags. The color transformation from RGB to CIE XYZ becomes a 1-D lookup in each channel followed by a 3 by 3 matrix transformation. The inverse conversion is a 3 by 3 matrix transformation followed by a 1-D lookup in each channel. The XYZ to RGB transform can be fitted into the LUT based operation of [matrix] -> [1-D LUTs] -> [multidimensional LUT] -> [1-D LUTs] by setting the multidimensional LUT and output 1-D LUTs to void operations (i.e., bypass these two operations). Due to the order of the operations, the inverse conversion (RGB to PCS) cannot be fitted into the LUT based operation. The good news is that this problem will be solved in the next version (version 4.x) of ICC specification. To implement the color transformation for this type of ICC profiles using the current ICC specification, it may be easier to define a different object/class. If any ICC profile is of this type, the floating linking process could be performed by a 3 by 3 matrix operation and three 1-D lookup processes (or 1-D linear interpolations). For integer linking, the 3 by 3 matrix operation could be performed by 1-D lookup operations and additions/subtractions.

3.2.6 A Smart CMM for the Next Generation of ICC Color Management

Gamut mapping is an essential step in cross-media color transformation. There are many gamut mapping techniques[4,13] to map out-of-gamut colors into the output device gamut. One approach is to simply clip out-of-gamut colors into the boundary of the output gamut. Because different out-of-gamut colors may be mapped to the same output color, relative relationships of different colors are changed. It has been known that this approach is not an idea solution for gamut mapping. Another approach is to adapt the input device gamut into the output device gamut or to soft-clip colors into the output gamut based on the relationship between the input and the output device gamuts. This method approximately reserves the relative relationship of different colors therefore the perceptual color quality is improved. Both input and output device gamuts must be known so that this kind of gamut mapping approaches can be applied. To optimize image quality, it seems that the input gamut should not have effect on color remapping as far as each pixel can be represented by the input device. Based on this logic, gamut mapping using the image gamut and the output device gamut properly should be an ideal solution for the perceptual rendering intent. However, the image-dependent gamut mapping must be performed for each image, while the gamut mapping using the input and the output device gamuts is updated only if the input profile or the output profile is changed.

The core functionalities of current CMMs is to link a set of ICC profiles and to perform color transformation from one color space to another. Because gamut mapping is a slow process and there is no sufficiently robust gamut mapping algorithms for general purposes, the gamut mapping cannot be performed in a CMM. The ICC (and Postscript) solution is to perform gamut mapping in the step to create ICC profiles (or CRDs). Because the input gamut (device gamut or image gamut) is not known while the gamut mapping is performed, gamut mapping cannot be optimized. This degrades the color transformation quality.

Another problem of the current ICC color management is the application of CMYK to CMYK printing in which the black channel is lost during the transformation[6]. In graphic arts market, there is a need to preserve K channel partially or completely. However, this is almost not available in the current ICC workflow.

To solve the gamut mapping problem in the current ICC workflow (and possible in the Postscript flow), an important function, gamut mapping, must be added into a CMM for real-time gamut mapping[7,8]. If a CMM constructs a gamut surface for the output device and a gamut surface for the input device (or the input image) successfully, it takes AToBi tag instead of gamut mapped data (BToAi tag) from the destination profile for gamut mapping and profile linking in a high-quality mode.

Here is an example for the conversion from sRGB monitor color space to a printer CMYK color space using the current ICC flow. A CMM takes an AToBi tag (a table for the input device to PCS conversion) from the first ICC profile and a BToAi tag (a table for PCS to the output device color space conversion) from the second ICC profile and links them together as shown in Fig. 1. The BToAi tag is a table (may includes a matrix, 1-D LUTs, and a multi-dimensional LUT) created by gamut mapping, GCR, and so on. As this tag is created, the profile creator does not know that the input color space is sRGB so the gamut mapping is not optimized for the mapping from sRGB to this device's CMYK.

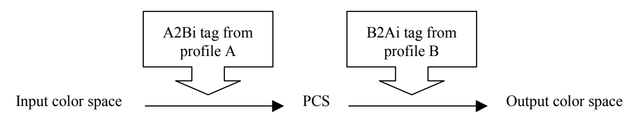

Figure 1. Color transformation in the current ICC color management

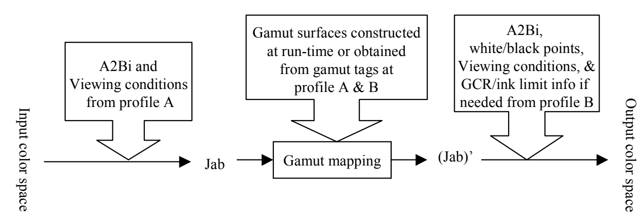

Figure 2. Color transformation by a smart CMM

A solution to this problem is to use the AToBi tags of both profiles, the viewing conditions of both profiles, media white points and black points of both profiles, and GCR information of the CMYK profile to perform gamut mapping (in this example, the gamut mapping is not image-dependent) and linking (BToAi tag is not used) as shown in Fig. 2. To have this function generally supported in a CMM, a new gamut surface tag may be added to the ICC profile format. If a gamut surface tag exists in each profile, it will be taken by the CMM for gamut mapping. Otherwise, the CMM creates a gamut surface structure for each device. If a gamut surface structure is not available in each ICC profile and the CMM fail to create one, the CMM switches to the current linking approach for profile linking.

To make this solution practical, we need to have a fast and robust gamut mapping algorithm implemented in the CMM. The computer processors are becoming more and more powerful, so the gamut mapping speed may not be a big issue. Furthermore, the color transformation lookup table is updated only if the input device or the output device is changed for non-image dependent gamut mappings. The procedures of the linking process to create a color transform table are: 1) to read the gamut surface tag from each ICC profile or to create a gamut surface for each device; 2) to convert input device color into PCS; 3) to adjust color based on viewing conditions and to convert color to a preferable color space for gamut mapping; 4) to convert color to the output device color space. If both the input and the output color spaces are CMYK (or CMYK plus additional channels) color space, the amount of the input K will be passed through PCS and be used as a reference value to determine the amount of the output K. Thus, the black preservation is achieved. This solves the black preservation problem existed in the current "dumb" CMM architecture. If the output is CMYK (or CMYK plus additional channels), the black generation and ink limit information (e.g. GCR) will be applied to the 4th step. If the gamut mapping is not image-dependent, this conversion object is updated only if one or both of the input and the output ICC profiles is changed.

A new gamut surface tag may be added to the ICC specification so that a CMM can use predefined gamut descriptions for faster gamut mapping. However, it must be defined carefully. For CMYK printers, the gamut changes with the change of ink limit and black usage, thus a gamut surface tag may be valid for just a set of ink limit and black usage condition.[9] If primary hues are not specified in gamut surface tag, another tag must be created to define the hues of primary ramps so that primary mapping can be performed for the saturation rendering intent.

CIE XYZ and CIE L*a*b* are two valid PCS color spaces in the current ICC specification. XYZ color space is not a luminance-chrominance color space and it is also not a uniform color space, so it is not a good color space for gamut mapping. L*a*b* color space has been known for its hue shift problem in the blue region for gamut mapping. Therefore both color spaces are not pertinent for gamut mapping. A new color space may be added to the PCS family or to replace the L*a*b* PCS for gamut mapping. CIE CAM97s Jab color space (a = C cos(h), b = C sin(h)) achieves better quality in gamut mapping than using CIE L*a*b* color space. It may be benefited to add this color space into the PCS family in the future for faster gamut mapping at real-time. IPT could be a good candidate for gamut mapping, but CIE CAM97s can be used for both color appearance adjustment and gamut mapping. However, CIE CAM97s color space exaggerates the chroma for dark colors. Because of this effect, the black point of a printer may be shifted far off the gray-axis. This may increase the difficulty for gamut mapping. CIE CAM97s may need modification so that chroma for dark colors is reduced.[10]

If a viewing conditions tag is not found in an ICC profile, the smart CMM applies default viewing conditions based on the type of the ICC profile (this information is in the profile header). If the black point is not available, a CMM will try to compute it based on the AToB1 tag, the white point tag, and viewing conditions (GCR information or other ink limit information may also be needed). Since a monitor (and RGB type printer) usually uses its full device range so its gamut surface can be constructed from other tags, a gamut surface tag is not required for this type of profiles. Some color spaces (e.g. e-sRGB, RIMM/ROMM RGB, L*a*b*) are served as encoding color spaces instead of physical color spaces, and their gamuts are not physical gamuts. Therefore we cannot use their full color space ranges for gamut surface construction. One method to prevent constructing a gamut surface for this type of profiles is to always create color space type ICC profiles for encoding color spaces and a CMM should not try to construct gamut surfaces for them. Another approach is that a gamut surface tag is required for all ICC profiles and a flag in the tag is used to specify whether a gamut surface can be constructed using its full input color space ranges. If both gamut surfaces are available from the profiles (or can be constructed), a smart CMM performs gamut mapping in real-time for the best quality mode. If the output device gamut is known, but the input device gamut is unknown, a CMM may provide another option to perform image-dependent gamut mapping. If the output device is a CMYK (or CMYK plus additional channels) printer, ink limit and GCR parameters are also required for the real-time gamut mapping and linking.[9]

For image-dependent gamut mapping, the gamut surface of the input image instead of the input device is constructed for gamut mapping. It may be difficult to achieve primary preservation for this type of gamut mapping. Therefore, image-dependent gamut mapping might not be good for the saturation rendering intent.

3.2.7 Quality/Speed Modes for a CMM

Five linking processes have been described. The approach described in the section 3.2.1 results in poor color accuracy therefore is not recommended for CMM implementation. The approach described in the section 3.2.2 achieves reasonable accuracy in most cases and the color transformation is fast, so it could be used in draft to normal modes. The approach described in the section 3.2.3 achieves high accuracy but the color conversion is slower than the method described in the section 3.2.2, so it could be used for a high quality mode. The method described in the section 3.2.4 requires many steps of transformation and the color conversion process is the slowest, thus it may not be practical for CMM implementation. The smart CMM linking provides an additional option for color transformation that can be enabled for the best-quality mode if the required information is available for smart linking.

The size of the multidimensional LUT created by the linking process also influences the color conversion accuracy. The larger the size, the higher the accuracy. Although the color conversion speed is not influenced by different LUT sizes after the linking is finished, it takes more memory and longer time to create a larger size of multidimensional LUT in the linking step. A smaller LUT could be used for draft to normal modes, and a larger LUT could be used for normal to best modes. The linking process could be performed in either floating-point operation or integer point operation. Linking by floating point operation may achieve higher accuracy and is also easier to implement. Linking by integer operation should be performed after data are scaled up to a higher bit depth so that desired precision could be achieved.

3.3 Gamut Warning

With the current ICC specification, the gamut tag is represented by LutType data structure, which cannot represent gamut surface accurately. It cannot be used to accurately predict whether a color is inside the gamut. To perform gamut warning more accurately in CIE L*a*b* color space, we can construct an L*a*b* to L*a*b* 3-D LUT by merging BToA1 and AToB1 tags and then setting a ΔE_{L*a*b*} threshold to decide a color is inside or outside the color gamut. This method should perform gamut checking more accurately than using the gamut tag. If the AToBi tag is the inversion of BToAi of the corresponding rendering intent for in-gamut colors, the gamut check for all rendering intents could be conducted by this approach.

4 Conclusions

We discussed how to accurately transfer color across a color system by focusing on the processing of the LutType (Lut8Type and Lut16Type) tags. The 1-D input and output LUTs are used to decouple the non-interdependent nonlinear factors of color transformation so that the multidimensional LUT can be used to approximate the multidimensional transformation more linearly. By applying 1-D LUTs properly, color transformation represented by LutType tags can be improved. If the 1-D LUTs are very nonlinear, Lut16Type should be used to achieve a higher accuracy. The example to create e-sRGB ICC profile shows the importance of representing channel-independent non-linear factors by 1-D input and output LUTs. The CIE L*a*b* is a uniform color space, therefore it is usually more suitable for serving as PCS for sampling grid points in the entire color space. However, the linear relationship between CIE XYZ and a typical monitor RGB color space makes it possible to sample the surface colors of a monitor color space by the grid points of the Lut16Type BToAi tag if CIE XYZ is used as PCS. This results in more accurate color transformation for surface and primary colors for a specific input monitor RGB color space, which is very important for graphic objects.

For CMM implementation, we proposed that Lut8Type data be converted into Lut16Type data at the linking step to simplify the CMM linking process. Instead of using the gamut tag, we proposed applying BToAi and AToBi tags to construct a more accurate gamut warning table at real-time. ICC linking through step-by-step sequence operations achieves fairly accurate color transformation and the transformation is fast, therefore it is recommended for draft to normal mode color conversion. Creating a deviceLink profile with an identity matrix for real-time color transformation improves accuracy but efficiency is lower, so it is recommended for normal to best mode color conversion. Step-by-step sequence operations through all ICC profiles using floating computation without linking achieves highest accuracy, which can be used for research, debugging, and profile re-generation and editing. Creating a smaller multidimensional LUT in the linking process requires less memory and the linking is faster, it can be applied to draft mode color conversion. Creating a large multidimensional LUT during the linking process improves the color conversion accuracy, so it can be applied to the color conversion for normal to best modes.

A smart CMM was designed to take an AToBi tag from the first ICC profile, an AToBi tag from the second ICC profile, optional gamut surface tags from both ICC profiles, the viewing conditions and media white/black points of both ICC profiles, and ink limit and black separation information of the output profile if needed to perform gamut mapping at linking time for high-quality color transformation. A new gamut surface tag may be created to specify whether a gamut surface can be constructed with the full range of the color space, and if the full range cannot be used, a gamut surface data set must be included. This will improve the reliability for the real-time gamut surface construction. A new color space may be added to the PCS family (or to replace L*a*b* space) to eliminate or to reduce the process for color transformation from PCS to another color space for real-time gamut mapping. Image-dependent gamut mapping can also be provided by a CMM so that gamut mapping can be performed in the case that the input device gamut is unknown or to improve gamut mapping quality. The black preservation is achieved for CMYK to CMYK transformation through this smart CMM.

In the coming revision of ICC specification, the Lut8Type and Lut16Type will be replaced by lutAtoBType and lutBtoAType. One simple approach for backward compatibility is to convert version 2 ICC profiles into ICC profiles based on the version 4.x specification, or to convert old tags into new tags in the CMM linking process. Thus, a CMM is not required for backward support the old tags for linking and color transformation. A CMM may also convert any RGB color space defined by colorant and TRC tags into LutAtoBType and LutBtoAType tags so that the linking and interpolation functions for LUT type tags can also be used for this kind of RGB color space.

References

1. Specification ICC.1:1998-09 File Format for Color Profiles, http://www.color.org/profiles.html/.
2. T. Kohler, "ICC Achievements and Challenges", http://www.color.org/.
3. M. Has and T. Newman, "Color management: Current Practice and the Adoption of a New Standard", http://www.color.org/.
4. J. Morovic and M. Luo, "The Fundamentals of Gamut Mapping: a Survey", J. Imaging Sci. Technol. 45, 283-290 (2001).
5. H. Zeng, "Profile Preservation in ICC Color Management", to be presented at Electronic Imaging Conference, January 2002.
6. H. Zeng, "CMYK Transformation with Black Preservation in Color Management System", to be presented at Electronic Imaging Conference, January 2002.
7. T. Newman, J. Haikin, and T. Kohler, "Supporting Both Preferred and Predictable Reproduction with ICC Profiles", Proc. IS&T/SIC 8th Color Imaging Conf., 65-69 (2000).
8. T. Kohler, "The Next Generation of Color Management System", Proc. IS&T/SIC 8th Color Imaging Conf., 61-64 (2000).
9. H. Zeng, "Ink Limit Control for Inkjet Printer Color Calibration", Proc. SPIE: Color Imaging: Device-Independent

Color, Color Hardcopy, and Graphic Arts VI, Vol. 4300, 93-96 (2001).
10. H. Zeng, "Gamut Mapping in a Composite Color Space", to be presented at NIP17: the 17th International Congress on Digital Printing Technology, September 2001.
11. PIMA 7667 WD1, Photography – Electronic still picture imaging - Extended sRGB color encoding – e-sRGB, http://www.pima.net/standards/it10/IT10_POW.htm#7667/.
12. M. Nielsen and M. Stokes, "The Creation of the sRGB ICC Profile", Proc. IS&T/SIC 6th Color Imaging Conf., 253-257 (1998).
13. C. Cui, "Gamut Mapping with Enhanced Chromaticness", IS&T/SIC 9th Color Imaging Conf., November, 2001.

Appendix: An Example for Accurate Color Representation Using a LUT Type Tag

This is an example showing the importance of using the input and the output 1-D LUTs in LUT type tags to improve color accuracy for ICC profiles.

There are two approaches to extend the gamut of a RGB color space. One is to move R, G, and B primaries toward the spectral locus. The other is to expand tone values to negative and to more than 100%. RIMM/ROMM RGB and Photoshop wide-gamut RGB use the first approach, while e-sRGB takes the second approach. The detail of e-sRGB definition can be found in Ref. 1. A brief description is shown here.

The sRGB primaries are adopted by e-sRGB. But the R, G, and B device values are extended to below 0 and more than 1 to expand color gamut. The conversion from CIE XYZ to linear e-sRGB is exactly the same as that for sRGB, which is

$$\begin{bmatrix} R_{e-sRGB} \\ G_{e-sRGB} \\ B_{e-sRGB} \end{bmatrix} = \begin{bmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (1)$$

If $R_{e\text{-}sRGB}$, $G_{e\text{-}sRGB}$, $B_{e\text{-}sRGB}$ < -0.0031308,

$$\begin{aligned} R'_{e-sRGB} &= -\left(1.055 \times \left(-R_{e-sRGB}\right)^{(1.0/2.4)} - 0.055\right) \\ G'_{e-sRGB} &= -\left(1.055 \times \left(-G_{e-sRGB}\right)^{(1.0/2.4)} - 0.055\right). \\ B'_{e-sRGB} &= -\left(1.055 \times \left(-B_{e-sRGB}\right)^{(1.0/2.4)} - 0.055\right) \end{aligned} \quad (2a)$$

If $-0.0031308 \le R_{e\text{-}sRGB}$, $G_{e\text{-}sRGB}$, $B_{e\text{-}sRGB} \le 0.0031308$,

$$\begin{aligned} R'_{e-sRGB} &= 12.92 \times R_{e-sRGB} \\ G'_{e-sRGB} &= 12.92 \times G_{e-sRGB}\,. \\ B'_{e-sRGB} &= 12.92 \times B_{e-sRGB} \end{aligned} \quad (2b)$$

If $R_{e\text{-}sRGB}$, $G_{e\text{-}sRGB}$, $B_{e\text{-}sRGB}$ > 0.0031308,

$$\begin{aligned} R_{e-sRGB} &= 1.055 \times R_{e-sRGB}{}^{(1.0/2.4)} - 0.055 \\ G'_{e-sRGB} &= 1.055 \times G_{e-sRGB}{}^{(1.0/2.4)} - 0.055\,. \\ B'_{e-sRGB} &= 1.055 \times B_{e-sRGB}{}^{(1.0/2.4)} - 0.055 \end{aligned} \quad (2c)$$

Finally,

$$\begin{aligned} R''_{e-sRGB} &= R'_{e-sRGB} \times 255.0 \times 2^{n-9} + offset \\ G''_{e-sRGB} &= G'_{e-sRGB} \times 255.0 \times 2^{n-9} + offset \\ B''_{e-sRGB} &= B'_{e-sRGB} \times 255.0 \times 2^{n-9} + offset \end{aligned} \quad (3)$$

where:

$$offset = 2^{n-2} + 2^{n-3} \quad (4)$$

and n is the number of bits used for each of the R, G, and B channels for encoding. For 10-bit, 12-bit, and 16-bit encoding, n is 10, 12, and 16, respectively.

To build an ICC profile, first we need to decide a color space for PCS. If L*a*b* is used, XYZ to L*a*b* nonlinear conversion must be included in the 3-D LUT of a LUT type tag with the version 2 profile format. 3-D interpolation errors will be introduced as a CMM uses it for color conversion. To avoid the interpolation error, we chose XYZ as PCS.

The e-sRGB specification defines the 3x3 matrix transformation from e-sRGB to CIE XYZ in D65 white point. Since the PCS is specified in D50 white point, a chromatic adaptation from D65 to D50 is performed to adjust the CIE XYZ value from D65 to D50. In order to create an e-sRGB ICC profile that is compatible with the official sRGB ICC profile created by Hewlett-Packard, we decided to apply the same white point adaptation formula, which is the Bradford transform without the non-linear portion. Thus the e-sRGB to PCS transform becomes[12]

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{D50} = \begin{pmatrix} 0.4361 & 0.3851 & 0.1431 \\ 0.2225 & 0.7169 & 0.0606 \\ 0.0139 & 0.0971 & 0.7141 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \quad (5)$$

Because e-sRGB extends RGB tone ranges to negative, the nonlinear transformations cannot be represented by Red, Green, and Blue TRC tags. Therefore, we cannot use R, G, and B TRC tags and R, G, and B colorant tags to represent the e-sRGB ICC profile as the sRGB profile created by Hewlett-Packard. The only approach to create a version 2 ICC profile for e-sRGB color space is to use LutType tag. We chose CIE XYZ color space as PCS to eliminate the non-linear transforms between CIE XYZ and L*a*b* color spaces. 3-D LUTs were created for the conversion between 16-bit e-sRGB and CIE XYZ color spaces. The goals are: to apply e-sRGB ICC profile to convert colors between e-sRGB color space and other color spaces with reasonable accuracy; and to convert colors from sRGB color space to e-sRGB color space and then back to sRGB color space with exactly the same RGB values in 8-bit accuracy.

To simplify the profile creation process, we created a 33x33x33 XYZ to e-sRGB LUT and a 33x33x33 e-sRGB to XYZ LUT for the BToA0 tag and the AToB0 tag, respectively. The 1-D input and output LUTs were set to identity curves, and the 3x3 matrix was also set to identity. With this e-sRGB ICC profile and the sRGB ICC profile, we tested the accuracy for sRGB → e-sRGB → sRGB conversion by Adobe Photoshop version 5.5. The procedures are: 1) to scale an sRGB image from 8-bit depth to 16-bit depth; 2) to convert it into 16-bit e-sRGB by Profile to Profile conversion using the Photoshop built-in color engine; 3) to convert it into 16-bit sRGB image; 4) finally to scale it down to 8-bit sRGB image. We found that there were large errors in a lot of colors. For example, an input green sRGB color (0, 255, 0) is converted back to sRGB values as (10, 255, 2).

Because most CMMs use trilinear or tetrahedral interpolation for 3-D interpolation, a nonlinear function represented by a 3-D LUT cannot accurately represent the nonlinear characteristics of off-grid points. Although the conversion between CIE XYZ and the linear e-sRGB is linear, the conversion between nonlinear e-sRGB and e-sRGB is nonlinear. Since the 3-D LUT for CIE XYZ to e-sRGB conversion includes the nonlinear transform, most of color transformation errors probably come from the nonlinearity. To reduce the 3-D interpolation error, we decided to use the input and the output 1-D LUTs to transform the nonlinear characteristics.

The current ICC profile specification does not allow offsets or negative values in 1-D LUTs or offsets in the 3 by 3 matrix of the LutType tags. To solve this problem, we shifted linear e-sRGB values so that they are non-negative and can be put into 1-D LUTs. An identity matrix is still used in the BToA0 and AToB0 tags.

To create the Lut16Type BToA0 tag, CIE XYZ values are converted into linear e-sRGB by inversing equation (5). Then e-sRGB values are added by 0.5 before scaling to guarantee almost all values are positive (any negative value is clipped to zero). The offset of 0.5 is subtracted from the 1-D output LUTs. Each of the three 1-D input LUT is a set of 1024 entries for identity transformation, and each of the three output 1-D LUT is a set of 4096 entries for the linear e-sRGB to the nonlinear e-sRGB transform with the offset of 0.5 so that the zero point is shifted to the 1024th node point.

In the Lut16Type AToB0 tag, each of the three input 1-D LUTs is created with 4096 entries for the transformation from non-linear e-sRGB to linear e-sRGB. 0.5 is added to the linear e-sRGB values in 1-D LUTs so that they can be represented by positive values (any negative value is clipped to zero). In the step to create 3-D LUTs, the linear e-sRGB is subtracted by 0.5 then converted to CIE XYZ by equation (5), and filled into the 33x33x33 3-D LUT. Each of the three output 1-D LUTs is a set of 1024 entries for identity transform.

We tested this e-sRGB ICC profile by Photoshop version 5.5 with the previous described procedure. Images were converted from sRGB color space to e-sRGB color space, and then back to sRGB color space. We found the output sRGB colors are exactly the same as the input sRGB colors in the 8-bit depth. This proves that using 1-D LUTs for nonlinear transformation and the 3-D LUT for linear transformation greatly improves the accuracy.

Because the output 1-D LUTs in the BToA0 tag and the input 1-D LUTs in the AToB0 tag are very non-linear, the color conversion accuracy is slightly decreased if these 1-D LUTs are reduce by half (2048 entries). The overall accuracy is also decrease if the size of 3-D LUTs is reduced to 17.