Unbounded Color Engines

Marti Maria; Little CMS; Palamós; Catalonia, Spain

Abstract

A Color Matching Method (CMM), also called a Color Engine, is a software component that does the color conversion calculations from one device's color space to another. This paper discuses a new working mode for CMMs that allows such software components to operate in a way that is not restricted by the gamut of the device encoding, the gamut of the profile connection space or any intermediate step. This allows ICC profiles to be used in new ways for a variety of applications. An open source CMM implementing this mode is also introduced.

Background

One of the main components of a color management system is the Color Matching Method, (CMM), which is the software engine in charge of controlling the color transformations that take place inside the system. By today, the vast majority of color management systems do use International Color Consortium (ICC) profiles.

ICC color management is based on device characterization profiles. Color transformations can be obtained by linking those profiles. This can be done because ICC has defined a standard profile connection space, the ICC PCS, which can either be XYZ or CIE L*a*b*. Each ICC profile describes how to do the conversion between the device color space and the profile connection space. In the ICC paradigm, all the smarts of the color conversion are embedded into profiles, and the CMM just concatenates them with some minor adjustments. This is known as the "smart profiles, dumb CMM" approach.

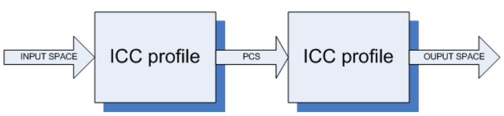

***Figure 1**. ICC color transforms as profile concatenation*

To implement the conversion device/PCS, ICC profiles can internally use three different math primitives: matrices, curves and multidimensional interpolation tables. Only certain combinations of those primitives are allowed, depending on the profile version and the direction being used.

Up to the ICC specification 4.2, profiles had a limit on the precision they could deliver. The internal encoding of profiles did force them to have a precision of 8 or 16 bits at most. That limit on precision was a stopper in the adoption of ICC profiles by some applications, like RAW photo processing or digital cinema, where greater levels of accuracy are required. See figure 2 for a DPX real world example, where input values 0-40 map to same output value 2, and 16-bit integer encoding cause severe quantization in shadows.

To overcome this limitation, in November 2006 the ICC approved the Floating Point Encoding Range addendum to the profile specification. With the introduction of floating-point data in the spec, ICC profiles are no longer limited to 8 or 16 bit, but to the broad range of 32-bit IEEE 754 floating point. That was a huge improvement when regarding precision and dynamic range, which is now only limited by floating point representation and can take as much as 10^{39}

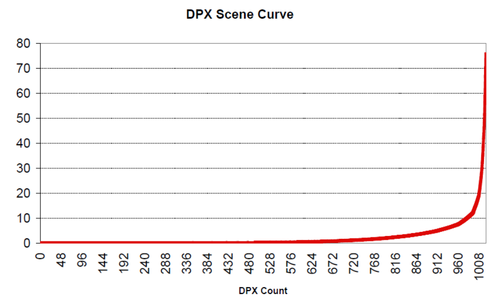

***Figure 2**. KODAK VISION2 500T Color Negative Film 5218 / 7218*

This addendum, however, introduced another improvement perhaps not so evident but equally important. Floating point encoding has a huge domain, which is nearly infinite, much larger than any real gamut. Based on that, one could think that a capable CMM using such kind of profiles could operate in an "unbounded mode" that would not have the limits that traditional CMMs have.

Further investigation has demonstrated that it is certainly possible to write such CMM. And moreover, some old, very popular, ICC profiles can be successfully used in this unbounded mode as well, without any modification at all. Compatibility with existing profiles is a key feature to properly leverage a new feature. Many of the yet-existing images are encoded in popular color spaces like sRGB or AdobeRGB, and have those profiles embedded. A CMM capable to work in unbounded mode with such images and with such profiles would certainly be easier to adopt that any system requiring new profiles and new encodings for every single image.

Bounds in ICC color management

In ICC specifications previous to the Floating Point Encoding Range addendum, device encoding range must have definite bounds. According that, most, if not all, of today's CMM perform some sort of clipping. To enumerate a few samples: Adobe ACE, Apple ColorSync and Windows ICM.

Clipping happens because several reasons. One is the encoding of the device space. If we use a 8 or 16 bit representation of the color space, all encodeable values are inside device gamut by definition, and there is no way to represent out of gamut values. For example, in the traditional 8-bit encoding, device values goes from 0 to 255. This is encoded in one byte taking all available bits, and therefore there is no way to represent negative numbers or values over 255. 16 bit gives more precision but still clips values to be inside realizable gamut.

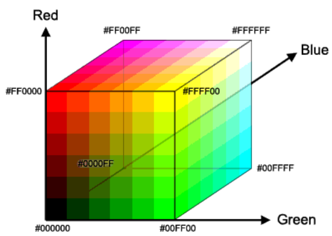

***Figure 3**. RGB device encoding bounding*

More subtle clipping can happen inside the CMM without any evident trace to the end user, as another source of clipping in ICC color management is the encoding of profile connection space itself. Version 2 Lab PCS is based on an ideal reflection print that has white point mapped on perfect diffuser and black point mapped on perfect absorber. Real device range is then scaled to this hypothetical media which has infinite dynamic range. As a result, real device black point is mapped to Lab (0, 0, 0), and real device white point is mapped to Lab (100, 0, 0). The mapping of the measured colorimetry of the device white to (100, 0, 0) is accomplished using linear XYZ scaling. The media white point tag is then used to undo the scaling of the device white to produce "ICC-absolute" colorimetry values which are relative to the assumed adapted white.

The reason to do this normalization is to maximize profile connectivity in perceptual and saturation rendering intents. In normal ICC operation mode, the device encoding bounding do not limit the dynamic range that can be supported because there are no restrictions on the relation between the device encoding minimum and black, or between the device encoding maximum and the assumed adapted white. But as a side effect of that mapping, values below black point are not encodeable; they result in negative L* or XYZ and most CMM clips them to zero. Highlights over white point may suffer same clipping, since the perceptual PCS does map device white point to Lab (100, 0, 0). Since this happens in the profile connection space, the end user does not see this effect directly but as an indirect result

XYZ PCS has an encodeable range of [0...1.99997] which encompasses most of today gamuts, It is unlikely that any device values will have corresponding D50 chromatically adapted XYZ values above 1.99997. However, it is possible for this to occur in some unusual circumstances, for example in situations of extreme fluorescence where the media white is much darker than some saturated colors. The media white is defined to be the lightest neutral color that a capture device can capture, or an output device can produce. It is also possible that some device values may have corresponding XYZ values that are negative. Such values can result from digital camera color analysis matrices, or chromatic adaptation transforms applied to extremely saturated blue colors. In most cases, it is acceptable to clip negative XYZ values to zero as such values do not correspond to real colors. However in some cases this may be unacceptable, for example if perfect round tripping is desired

Lab PCS is more limited. A device value of (0, 255, 0) in AdobeRGB, for example, results in Lab (83.2, -128.1, 86.1) which is not encodeable in Lab PCS. Lab PCS has *a*/b** axis restricted to $(-128 \leq ab \leq 127)$ in version 4 of the ICC spec. Version 2 of ICC spec have slightly different limits: $(-128 \leq ab \leq 127.996)$ but again that's not enough to hold entire AdobeRGB gamut.

Bounding CMM example

Let's take an example to demonstrate the PCS clipping effect by using Adobe Photoshop CS4. If we setup Photoshop to use sRGB as working space, and convert by means of relative colorimetric intent a Lab value of:

Lab = (0, -120, 0)

We will obtain *sRGB = (0, 0, 0)*. That seems to be perfectly reasonable as L*=0 maps to black. Now we can try a symmetrical value, but on b* axis

Lab = (0, 0, -120)

In this case we obtain sRGB = (0, 27, 182), which is somehow surprising given the (0, 0, 0) result obtained with the previous value. Why Photoshop does clip one axis (a*) and does not clip the other (b*) in the same fashion?

The reason of such values is clipping performed on the PCS. If we take first Lab value, and convert it to XYZ, we obtain;

XYZ = (-2.9, 0, 0)

Which has a negative value on X colorant. Since the CMM performs PCS clipping, the XYZ value becomes (0, 0, 0) which is readily translated to sRGB = (0, 0, 0). On the other hand, Lab = (0, 0, -120) converted to XYZ results in (0, 0, 33.1), which does not suffer from PCS clipping, and therefore is translated to (0, 27,182) by the sRGB equations:

$$\begin{bmatrix} R_{\text{linear}} \\ G_{\text{linear}} \\ B_{\text{linear}} \end{bmatrix} = \begin{bmatrix} 3.2410 & -1.5374 & -0.4986 \\ -0.9692 & 1.8760 & 0.0416 \\ 0.0556 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$C_{\text{srgb}} = \begin{cases} 12.92 C_{\text{linear}}, & C_{\text{linear}} \leq 0.0031308 \\ (1+a) C_{\text{linear}}^{1/2.4} - a, & C_{\text{linear}} > 0.0031308 \end{cases}$$

***Figure 4**. XYZ to sRGB conversion*

If we inspect sRGB equations, there is no place where hard clipping is needed, and the math could perfectly deal with negative XYZ numbers, despite those may have no physical meaning at all. Computing the conversion without any clipping does result in following values:

(0, 0, -120) → (-536.1, 27.2, 181.7)
(0, -120, 0) → (-307.0, 47.6 -7.0)

Those are clearly out of sRGB gamut. At that point we may choose to perform the clipping as usual or to keep negative numbers and let the calling application to take the pertinent action on depending on the use application has for the resulting colors. That latter would be an example of a CMM working in *unbounded mode.*

Unbounded mode

Most CMM does allow using a variety of formats for both input and outputting raster data. Some CMM allow using floating point as an additional format. Since floating point allows a wide range of values, the bounding introduced by the device encoding format no longer applies. CMM often clips those floating point values to be in [0...1.0] range because convention. But at that point one could argue that this bounding is not strictly needed. Some profiles are built by using elements that does not impose any bounds at all. If a color transform is setup by using exclusively this kind of profiles, and floating point is used as input and output format, there are no constrains and the color transform is not limited by any bounds. We would name this special mode as *unbounded CMM.*

Some colorimetric space conversions are also likely to work unbounded. Conversion from Lab to XYZ and vice-versa can run unbounded, despite that would yield colors that are not physically realizable. ICC currently only supports XYZ and Lab as profile connection spaces.

$XYZ \rightarrow Lab$

$$a^* = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right]$$
$$b^* = 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right]$$

with

$$f(r) = r^{1/3} \qquad , r > 0.008856$$
$$f(r) = 7.787r + 16/116 \qquad , r \leq 0.008856$$

$Y \rightarrow L$

$$L^* = 116\left(\frac{Y}{Y_n}\right)^{1/3} - 16 \qquad , Y/Y_n > 0.008856$$
$$L^* = 903.3\left(\frac{Y}{Y_n}\right) \qquad , Y/Y_n \leq 0.008856$$

Figure 5*. XYZ to CIE L*a*b* conversion*

At that point we should note that ICC profile connection space, on the perceptual intent, does use CIE L*a*b in a way that does not represent physical colors but a scaling of the device gamut. Black point in Version 4 perceptual PCS is represented as (3.1372, 0, 0) in Lab coordinates. In Version 2 PCS, Black point is represented by a perfect absorber (0, 0, 0). Because that, some situations can end in negative XYZ or L* numbers, or values of L* that exceeds 100.

An unbounded CMM should keep all those temporary values, and clip only in situations where there are no other options. Indexing tables is a clear example, since the table imposes an input domain, values outside this domain should be clipped as there are no table entries for other values and the unbounded function is undefined. In those situations, the CMM may decide to switch back to bounded mode.

Version 4 does introduce parametric curves. On certain parameters, some of those curves may be undefined on negative domains, or return complex numbers. In that case the CMM should also switch back to bounded mode. For most cases, on parametric curve types, as well as in pure exponentials already found on V2, unbounded mode can be used without problems.

Use cases

Unbounded mode opens ICC color management to several new applications. First use to come in mind is image processing. Many image processing operators can be effectively used on data which is negative, or above white. Other area where unbounded mode makes a lot of sense is in gamut mapping, where pivot points may need to be computed in the target color space but outside the device gamut. In general, all situations where a perfect round-tripping is needed can benefit from unbounded mode.

Some editing is best done in a linear-light Scene Space Effects (such as motion blur or adding shadows) are more photorealistic when made in scene colorimetry.

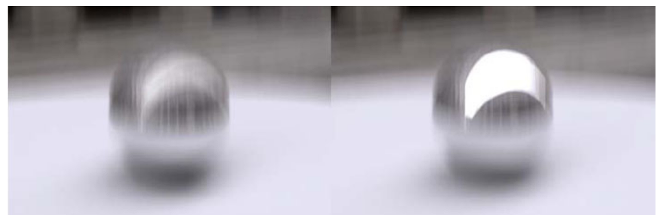

Figure 6*. Motion blur in linear scene color space versus same effect in a clipped space.*

Another use would be workspaces. By using unbounded mode, an AdobeRGB image can be manipulated on sRGB working space without any loss. Preview of unbounded sRGB to monitor space is possible and easy, as negative or out of sRGB gamut may fall inside monitor or printer gamut.

There are other not so evident uses for unbounded data, like high dynamic range imaging. Since unbounded mode can naturally deal with highlights (L* over 100) and drop shadows (values below black point), unbounded mode can be used on motion cinema, where dynamic range is often broader that the reference color space. Note that adjustments for viewing conditions can result in such out of range values. Even for image storage, unbounded mode may be interesting. If we want to store a bunch of images in a common colorspace, but we want to avoid data loss, a format capable to store out of gamut values (like float TIFF, for example) and an unbounded CMM may be very handy.

Those are only samples of the amount of new applications unbounded modes would have. It is likely further investigation will find new and exciting ways to use such feature. An unbounded CMM can be turned into a bounded one. In this case, clipping happens only on the last stage or when is absolutely required.

Suitable profiles

A big number of yet-existing profiles have been inspected by the author in order to know how well unbounded mode could be used. Some of yet-existing V2 ICC profiles are already not limited by bounds. That is the case of profiles implemented as a 3x3 matrix plus a set of curves. Those are known as “matrix-shaper” profiles. Well known samples are AdobeRGB1998 and sRGB profiles. AdobeRGB does use a pure exponential for curves, so the profile may be directly used in unbounded mode. The traditional sRGB does use tables for describing the curve, and unfortunately this prevents the profile to be used in unbounded mode. However, if we examine the sRGB specification, the curves are defined in all real domains, and using a V4 profile we can implement an unbounded sRGB by using a parametric curve of type 4: (IEC 61966-2.1)

$$y = f_4(x) = cx + f, \qquad 0 \le x < d$$
$$= (ax + b)^{\gamma} + e, \qquad d \le x \le 1$$

Where:

$$\gamma = 2.4;\ a = 1.\,/\,1.055;\ b = 0.055\,/\,1.055;$$
$$c = 1.\,/\,12.92;\ d = 0.04045;\quad e = f = 0; \tag{1}$$

Using this mechanism, a built-in version 4 sRGB profile has been included in the reference implementation, see below. Other important profiles found to work well in unbounded mode are version 4 profiles for synthetic color spaces like ISO 22028 2 ROMM RGB and RIMM RGB (Pro Photo RGB). Eci RGB V2, the working colour space profile recommended by ECI, has been also found to work flawless, as well as many others not mentioned here because the limited extension of this paper.

CMM Implementation and availability

As a practical demonstration that unbounded mode CMM can be implemented and to experiment with this new mode, ***Little CMS***, a very popular open-source CMM has been accommodated to work in unbounded mode. Implementation has been very difficult. It toke about 2 years, and changes are significant: all computation is now done in floating point and clipping, if needed, happens late on the pipeline. Version 2.0 is practically a full rewrite of the original color engine. Because it is no longer backwards compatible, a major version bump has been needed, so Little CMS 2.0 is now available for free as one of the very first unbounded CMM at any cost. The license used is MIT, which is very liberal:

http://www.opensource.org/licenses/mit-license.php

The full package is available at:

http://www.littlecms.com/downloads.htm

All internal computations have been rewritten to use floating point and all bounding and internal clipping operations have been removed. This had a severe impact on performance, so a special optimization module for bounded formats was created. Now the color engine does compute always the unbounded transform and passes it to the optimizer. The optimizer, after inspection of the input and output formats, is able to clip the data if bounded mode is detected. Despite what it may seem, that accomplished a performance gain of about 400% in some cases, like 8-bit RGB to RGB transforms.

To check the engine capabilities, several independent client programs have been developed. ***Transicc*** is a command-line utility that can process plain text or CGATS files containing data across chains of ICC profiles in either bounded or unbounded mode. ***Jpgicc*** and ***Tificc*** allow applying ICC transforms to TIFF and JPEG files. There is also a *Matlab* MEX, which is capable to apply Little CMS 2 color transformations to images and arrays using the *Matlab* package.

Conclusions

Although unbounded mode implementation has its challenges, it is certainly a valuable tool. This paper describes the effects bounding has, compare results against unbounded mode, identifies some of the advantages unbounded mode may have, and introduces an open source implementation under a very liberal license. Finally, some of the possible uses for unbounded mode CMM are outlined.

References

[1] Specification *ICC.1:2001-04 File Format for Color Profiles*, International Color Consortium, available at http://www.color.org/

[2] Specification “ICC.1: 1998-09, File Format for Color Profiles,” International Color Consortium, available at http://www.color.org/

[3] M. Maria. Little CMS Engine – How to use the Engine in Your Application, 2004, Available from http://www.littlecms.com

[4] A Standard Default Color Space for the Internet: sRGB, Version 1.10, November 5, 1996. ICC

[5] Eastman Kodak Company, “Reference output Medium Metric RGB Color Space (ROMM RGB) White Paper,” Version 2.2, 1 July 1999.

[6] R. W. G. Hunt. Measuring Colour. 8Fountain Press, 3rd edition, 1998.

[7] Mary Nielsen and Michael Stokes, The Creation of the sRGB ICC Profile. IS&T/SID Sixth Color Imaging Conference: Color Science, Systems and Applications. Scottsdale, Arizona, 253-257 (November 1998)

[8] Adobe RGB (1998) color image encoding. Version 2005-05. Available at http://www.adobe.com/digitalimag/adobergb.html

[9] ICC Floating-Point Device Encoding Range, International Color Consortium, available at http://www.color.org/ICCSpecRevision_02_11_06_Float.pdf

Author Biography

Marti Maria is a color engineer at the large format printer division of Hewlett-Packard. He worked previously at ICR; a company specialized in imaging and color. Marti is also the author of well-known open source color oriented packages, like the LittleCMS open CMM and the LPROF profiler construction set. He has contributed to several color books and was session chair on Color & Imaging Conference 16.