Usage of prelinearization stage on color-management application-specific integrated circuit (ASIC)

Marti Maria, HP
marti.maria@hp.com
ICD Barcelona

Abstract

This paper presents a method to compute a linearization stage that may increase accuracy on lookup based RGB to RGB color transforms as implemented in ASICs. Such method is especially effective when apparent gamma of input and output RGB spaces are quite different. The effect of the linearization stage in RGB to RGB color transforms is evaluated theoretically and experimentally though psychophysical testing using real images. Both theory and the test results confirm a significant performance increase in terms of image quality when linearization stage is being used.

Introduction

When dealing with RGB to RGB color transforms, many implementations of color-management ASIC rely on 3D look up table (LUT) with interpolation. Yet, although the result of 3D LUT is in general accurate, obtaining proper accuracy when source and destination spaces are very different requires a high number of grid nodes. ASIC devices usually have severe constraints on resources, so alternative solutions should be considered.

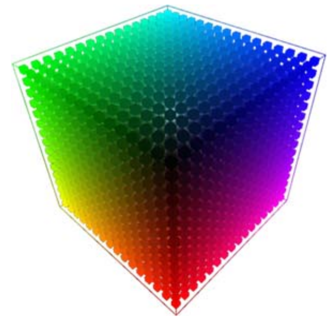
This paper describes a method that has proven to be effective on increasing resolution on RGB to RGB transforms implemented as 3D look up table. This method involves a previous linearization stage which is found in some latest generation color-management ASICs. The algorithm relies on three basic premises that are true on many common RGB spaces:

- R=G=B represents a neutral value respect to space's white point
- Gamma of each component is approximately decoupled of other components.
- Both RGB spaces are smooth

Despite the algorithm is not expected to work on RGB spaces out of those requirements, it has been found to perform well if characteristics are close enough. That is, an RGB space on which gamma is not completely decoupled, like those offered by cheap LCD monitors, may also work if the interaction of channels is not strong close to neutral axis. In any case, a non-suitable conversion is detected by the algorithm and whole linearization stage is then discarded.

3D look up table with interpolation

The three-dimensional lookup table (LUT) with interpolation is one of the most active research areas in color technology. Several applications of 3D interpolation in scanner, monitor and printer calibration have been already reported in the literature. It consists of three parts—packing (or partition), extraction (or find), and interpolation (or computation)[1]



Packing is a process that divides the domain of the source space and populates it with sample points to build the lookup table. Non-lattice points are then interpolated by using the nearest lattice points. This is the step where the extraction performs a search to select the lattice points necessary for computing the destination specification of the input point. The last step is the interpolation where the input signals and the extracted lattice points are used to calculate the destination color specifications for the input point. There are four geometrical interpolations — trilinear, prism, pyramid, tetrahedral — and many variations.

Due to the low computation and implementation costs, fast processing speed and a simpler analytical scheme to obtain the inverse lookup table, the tetrahedral interpolation has generated a lot of interest. This method has been applied successfully to device characterizations and calibrations, and color renditions of electronic images.

The first 3D interpolation that appeared in literature is the trilinear interpolation, disclosed in 1974 British patent by Pugsley.[2] The application of tetrahedral interpolation to color space transformation was latter patented by Sakamoto and Itooka in US patent 4,275,413 (1981). Both patents have expired.

Compared to other transformation techniques, the 3D LUT approach provides higher accuracy because it has the advantage that the color space can always be further divided by increasing the sampling density. If the sampling rate is high enough, the linear interpolation becomes a very good approximation for computing the point that is not a lattice point.

Apparent gamma

Many RGB color spaces are non-linear on respect to CIEXYZ colorimetric space. However, it is quite frequent this non-linearity has channel independence. In this way, a RGB color space may be described as a non-liner tone curve, which takes place at each R, G and B channels, and then an approximately linear part, with a 3x3 matrix multiplication to transform from one primary system to another. This is certainly the case of display monitors and synthetic workspaces like sRGB, Adobe RGB or Apple RGB. Other RGB spaces, like RAW on digital cameras and native device space for some printers have been accommodated to those constrains just because that makes such spaces easier to manipulate.

One of the first devices to use natively RGB was CRT displays. Traditionally, the CRT response curve is plotted as a power function, and the power is called gamma. One common model for CRT displays is the Gamma-gain-offset model, described by Berns[3].

$$R_{Scalar} = \left[gain_r \left(\frac{dc_r}{dc_{r,\max}} \right) + offset \right]^{\gamma}$$

Because this heritage, we could therefore compare any RGB space with a CRT analog, and talk about "apparent gamma", which would be the exponent used if the RGB space would be fitted on a CRT model. Certainly most parameters would make no sense at all on many RGB spaces, because the true nature of the involved device does not correspond to a CRT display, but this fitted gamma would still give important information about the order of the space. In this way, we could assign an apparent gamma 1.0 to CIEXYZ and an apparent gamma of 2.4 to CIE L*a*b*

According to a recent study, CIEXYZ values can be obtained for each RGB channel by interpolating the tristimulus values measured for the corresponding primary RGB values, plus the channel-interdependency assumption can be used to predict the CIE tristimulus values for any arbitrary combination of RGB values.[4]

The problem

When using 3D lookup table with interpolation, these nonlinear transformations have a profound influence on the precision requirements, because they modify the relative spacing of quantization levels. If input RGB is quantized to a number of equally spaced perceptual levels, these will be mapped to unequally spaced perceptual levels by gamma correction; some of those levels, particularly in the highlights, may be so closely spaced that they are imperceptible.

The issue comes when apparent gamma of both color spaces is quite different and the number of available grid nodes is small. This is readily evident when transforming a continuous gray gradient across 3D LUTS of 6 and 17 grid points. Let's take for example a RGB to RGB transform going from camera RAW (apparent gamma 1.0) to sRGB (apparent gamma 2.2)

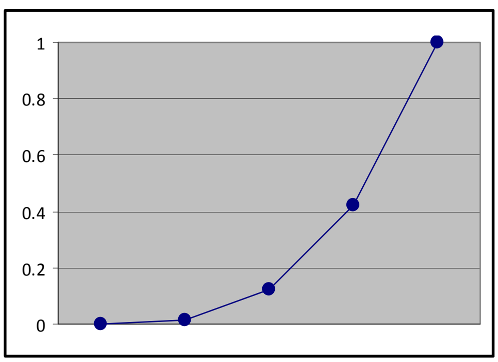


Figure 1

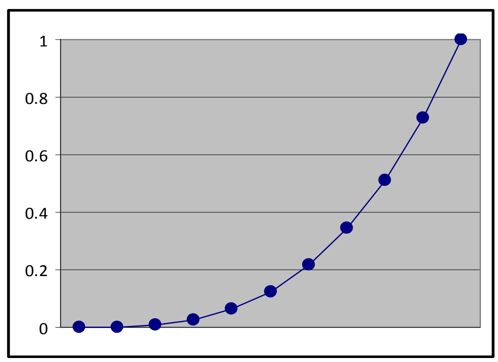


Figure 2

On figure 1, there is a representation of the quantization effects generated by a 6 point 3D look up table. On increasing the number of grid points to 17 (fig .2), the exponential curve becomes smoothed. This is very evident on images as blocking and loss of shadow detail (fig. 3, fig. 4).

Figure 3

Figure 4

Proposed solution

One possible solution to this problem is to use non-regular sampling, as that would assign more control points in zones where more resolution is required. A way to implement that is to use linearization curves.

Let's assume we have function $\boldsymbol{f}$ describing the *RGB* → *RGB'* color transform. This function is not mean to be computationally efficient and may be implemented by using device models, regression or any other methodology. We could therefore, evaluate any RGB value across this transform and obtain the corresponding RGB' triplet. With the obtained points we can then build the linearization curve plus the 3D LUT.

The proposed method includes three steps. First step builds 3 linearization curves. Second step validates those curves, and third step computes values for each lattice point on the 3D grid.

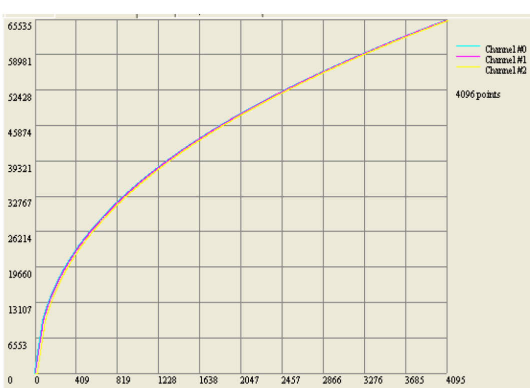
Tone curve construction.
For obtaining the tone curve for each RGB component, a number of RGB triplets are evaluated across the color transform. In 8-bit RGB transforms, 256 values are used. In 16-bits color transforms, a maximum of 4096 values was found to be enough.

The input values are meant to be near neutral, with (R=G=B). The obtained values are not necessarily neutral as R, G and B components may differ. This generates 3 sets of values ranking from zero to white point.

Those points are then fitted and smoothed into curves by using finite difference smoother[5]. The algorithm for finite difference smoother was published as early as 1923 by E.T. Whittaker[6] and is very well suited to discrete smoothing and interpolation. A combined measure of fit and smoothness is:

$$S = \sum_{i=1}^{m} w_i (y_i - z_i)^2 + \lambda \sum_{i=n+1}^{m} \left(\sum_{j=0}^{n} a_j z_{i-j} \right)^2$$

Where lambda is a parameter by which we can trade smoothness of $\boldsymbol{z}$ against fit to data $\boldsymbol{y}$. Some of the data points $\boldsymbol{y_i}$ may be missing, and are handled across weight $\boldsymbol{w_i}$. Normally $\boldsymbol{w_i} = 1$, but when $\boldsymbol{y_i}$ is missing, the corresponding $\boldsymbol{w_i} = 0$ and we can give any value we like to $\boldsymbol{y}$. To minimize $\boldsymbol{S}$, we set partial derivatives $\boldsymbol{\delta S / \delta z_i}$ equal to zero and obtain a system of $\boldsymbol{m}$ equations. A simple adaptation of Cholesky decomposition[7] is used to solve the system. In order to assure endpoints, values for black and white are fixed in 0 and 255, so the smoother is not allowed to move those points.



Next, the lowest part of the curve is replaced by a slope-limiting linear part in order to get rid of noise and high slope situations. The cutoff is placed at 2% of the curve, which only affects digital counts 0...6 in 8 bit. This linear tram has proven to be very effective on zones where monotonicity is hard to obtain.

Tone curve validation

It is very important to have a way to automatically validate the tone curves, as wrong curves may be very destructive on terms of accuracy, and may introduce unwanted artifacts like blocking or contouring. Some checks are performed on the obtained curves, if they fail, a substitution mechanism would take place. The tests performed are:

- Check for identity curves. Each point of the curve is compared against an identity line. If most values (80%) fall within a range of 1-3 contone values of identity, then whole tone curve is discarded as it would have no effect on final transform. Slope-limiting step is not checked on this test.
- Check for non-monotonic curves. Monotonicity is required in almost any kind of color transforms. Slightly non-monotonic curves are possible due to smoothing artifacts, so it is quite important to get rid of those parts. Non-monotonic curves are not reversible and do not work with this algorithm.
- Check for Endpoints. Curve should begin at zero and end at pure white, 255 on 8-bit and 65535 on 16-bit. While failure to do so is not necessarily an error state, it may be an indication of other kind of discrepancies.

If any of those tests fails, a backup strategy is used. A statistical check is then performed on original set of points to infer whatever a simple exponential may be used. The method involves computing the average and standard deviation of ***log(y)/log(x)*** being ***y*** the value obtained on the tone curve and ***x*** the value used as input. Lower 7% is ignored for same reasons of slope-limiting. If standard deviation is < 0.7, then the curve is meant to be approximately an exponential, and then a pure exponential (with slope-limiting) is being used instead.

If neither method succeeds, then the RGB to RGB color transform is not suitable to be optimized by means of tone curve and is up to the CMM to use a different strategy, which may be to use more grid points for 3D LUT or just mark the whole transform as non optimizeable.

Populate the 3D grid

Third and last step involves the packaging part. In order to compute the values for each node, the tone curves are inverted. Since step 2 has assured monotonicity, reversing curves is an easy task. In the case of exponential fitting, the curve may be analytically inverted in both trams, slope-limiting and exponential parts. A more general case may be handled by a simple binary search, since those curves are quantized to 8 or 16 bits. Whole RGB input space is then sampled at regular intervals. Those intervals conform the indexing of RGB input space. Each RGB triplet is then linearly interpolated across inverse tone curves and the resulting values are evaluated across color transform. Obtained RGB is used to populate the lattice point.

Experimental Results.

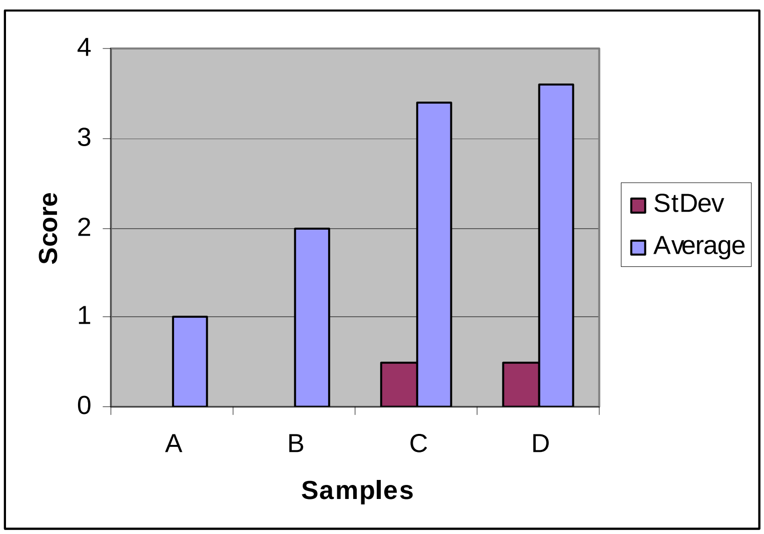
To test the performance of the proposed approach, a real-world example was chosen. The test case is a conversion Camera RAW → sRGB. Camera RAW is a preferred format for most high-end photographers. It has certainly some advantages to other formats, but the apparent gamma of this space is about 1.0 and this makes difficult to convert it to a more perceptually uniform space like sRGB, which has an apparent gamma of 2.2. The mismatch between gamma values makes the conversion hard when only a 3D LUT is available. Without linearization, 33 or more points per dimension would be needed in order to avoid artifacts.

Four test images were generated by the author using by 6 using, and 17 grid points, with and without linearization stage:

A. 6 points without linearization
B. 17 points without linearization
C. 6 points with linearization
D. 17 points with linearization

Images were stored and manipulated at 16 bits to avoid quantization effects. The color transform evaluated was setup using ICC profiles with far more precision that the test case. Final images were printed on a HP Design Jet Z3100, which is a large format printer aimed to photographer market. Media used was Instant-dry photo satin paper. The 3DLUT was computed by using the littlecms color library.

Five high-trained observers and two non-professionals were asked to evaluate the obtained plots by sorting them by preference, putting the plots they like more fist and those plots with visible defects or with artifacts last. This procedure effectively rates each plot in a scale of 1-4. Average score for each plot is shown in a bar diagram. The Standard deviation is also depicted.



All observers agreed 6 point LUT without linearization performed worst, followed by 17 points without linearization. It is also remarkable to note that many observers perceived both plots using linearization 3) and 4) as identical.

Conclusions

In this paper we have presented a method to compute a linearization stage that may increase accuracy on lookup based RGB to RGB color transforms on both 8 and 16 bits. This method is especially effective when apparent gamma of input and output RGB spaces are quite different. A psychophysical testing involving both trained and untrained observers have confirmed it greatly improves the accuracy of transforms on real images.

Author Biography

Marti Maria is a color engineer in the imaging and color group at the large format printer division of Hewlett-Packard. He has as degree in telecommunication engineering from the Universitat Politécnica de Barcelona. His research interests span various color imaging technologies and techniques for computer based graphics.

---

[1] Henry R. Kang. *Color Technology for Electronic Imaging Devices*. SPIE Optical Engineering Press, 1997.
[2] P.C. Pugsley, "Image reproduction methods and apparatus", British patent 1,369,702 (1974)
[3] Gamma function (GOG) (Berns, Roy S. Berns, Methods for Characterizing CRT displays, Displays 16, 173-182 (1996))
[4](G. Sharma, "LCD versus CRTs color calibration and gamut considerations", proceeding of the IEEE, vol 90, no. 4, pp. 605-622, April 2002)
[5] Eilers, P.H.C. (1994) Smoothing and interpolation with finite differences. Graphic Gems IV, Heckbert, P.S. (ed.), Academic press.
[6] (Whittaker 1923) E.T. Whittaker. On a new method of graduation. Proceedings of the Edinburgh Mathematical Society. 41:63-75, 1923.
[7] Roger A. Horn and Charles R. Johnson. *Matrix Analysis,* Section 7.2. Cambridge University Press, 1985